Social Shuffle

Music Discovery with Tag Navigation and Social Diffusion

Doctoral Dissertation submitted to the

Faculty of Informatics of the *Università della Svizzera Italiana*

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

presented by

Cédric S. Mesnage

under the supervision of

Prof. Mehdi Jazayeri

March 2012

Dissertation Committee

Prof. Fabio Crestani	Università della Svizzera Italiana, Switzerland
Prof. Marc Langheinrich	Università della Svizzera Italiana, Switzerland
Prof. Joemon Jose	University of Glasgow, Scotland, UK
Prof. Yngve Sundblad	KTH Royal Institute Of Technology, Stockholm

Dissertation accepted on 7 March 2012

Prof. Mehdi Jazayeri
Research Advisor
Università della Svizzera Italiana, Switzerland

Prof. Antonio Carzaniga
PhD Program Director

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Cédric S. Mesnage
Lugano, 7 March 2012

À mes parents.

(31:11) intellectualia, et ex iis operibus alia instrumenta seu potestatem ulterius investigandi; et sic gradatim pergit, donec sapientiae culmen attingat.

Tractatus de Intellectus Emendatione
Baruch de Spinoza

Abstract

This thesis tackles the problem of discovering music for users in a social network, introducing the concept of social shuffle and its implementation as a live experiment in social based recommendation, Starnet, and show that recommendations based on a user's social network is strongly effective in introducing a user to new music that she enjoys.

I investigate the generation of tag clouds using Bayesian models and test the hypothesis that social network information is better than overall popularity for ranking new and relevant information. I propose three tag cloud generation models based on popularity, topics and social structure. I conducted two user evaluations to compare the models for search and recommendation of music with social network data gathered from Last.fm. Our survey shows that search with tag clouds is not practical whereas recommendation is promising. I report statistical results and compare the performance of the models in generating tag clouds that lead users to discover songs that they liked and were new to them. I find statistically significant evidence at 5% confidence level that the topic and social models outperform the popular model.

I report on an experiment on social diffusion for music discovery. I describe the experimental methodology which includes the making of a music videos dataset and the creation of a social application. I give a statistical analysis of the participants ratings which shows that social diffusion leads to more good recommendations.

I conclude and show that the social shuffle is an effective mechanism for information recommendation and that social relationships are relevant data to enhance information navigation.

Acknowledgements

I am deeply thankful to my family and friends.

Contents

Figures

Tables

Chapter 1

Introduction

> "Music recommendation is more an Art than a Science."
> In "The Dark Art: Is Music Recommendation Science a Science?"
> Michael Papish.
> First workshop on Music Recommendation and Discovery
> (WOMRAD) Keynote. Sep 2010.

How do you discover new music? You can listen to the radio, watch TV or read music magazines. Most of the time you discover music randomly, by walking into a concert or a pub where they have their own music library. Another powerful mechanism is social recommendation, when friends recommend each other music or themes they discovered. If a track is recommended to someone by a friend, and she recommends it to someone else, this is social diffusion. The World Wide Web and the emergence of online social networks give us an opportunity to build software support for random selection and social diffusion. In this thesis I present my work in comparing social diffusion and other exploration processes, using tag data and ratings from existing social networks in live experiments.

1.1 Research questions

This thesis focuses on building tools to discover information on the World Wide Web, particularly I chose to experiment on discovering music. My interest is specifically on discoveries that are serendipitous. Most findings in every day life as well as in scientific life are serendipitous in nature, *i.e.* the person is put on a research track and make a happy discovery by accident. For instance Louis Pasteur discovered pasteurization while looking for the essence of matter and Christopher Colombus discovered the american continent whereas he was looking for a new navigation route to reach India. In this section I argue why the current mechanisms available on the Web fail to facilitate serendipitous information discovery.

Search vs discover The most used tool currently on the World Wide Web is search engines. Search engines are simple in interface, they propose the user with a text box in which the user can enter a text query. The engine finds relevant web resources to the query and presents the user a sorted list of resources. This is very practical when you know what you look for but rather unuseful if what you want is to discover something you do not know about.

Hypertext = Manual discovery The web is built on the concept of hypertext. The idea that a web resource is identified by a url (uniform resource locator) and web pages can include links to other web pages by pointing an anchor to a different url. This allows for web browsing, going from a page to another by following links. This is a major activity of people on the web, going to a page and wandering through web pages learning more about a topic and discovering new ones. This process is heavy on the user as it requires her to figure out which link to click on or abstracting what the page is about to start a new search as opposed to recommendation where the system computes what is the next item to be seen by the user.

Recommenders long tail Recommender systems enable to present the user automatically with resources of interest by reasoning on other users behavior on the site. For instance Amazon proposes related items with the famous "Customers who bought this also bought...". The problem with this type of recommendation is that the few most popular items always get recommended. The task of a recommender system is to drive people towards the long tail items[12], items that are not popular but might be relevant. Collaborative filtering is the process of recommending resources of interest between users who have similar taste. The recommendations are consequently within the scope of their taste, if they discover new information it is on topics that are within their defined tastes instead of helping them opening and discovering their taste and therefore new items of interest on topics they might not know about.

In this thesis I chose to experiment on the discovery of music as an application field, but the ideas apply to other kinds of data as well.

1.2 My approach

The World Wide Web evolves, and it evolves fast. Recently the web as seen the apparition of various artifacts, often named under the ombrella of Web 2.0, that are relevant to explore to enhance discovery.

Social network software With more people connected to the internet, the interest grew in finding friends and keeping track of what they are up to. Facebook is certainly the most successfull of all. An interesting aspect of Facebook is that it offers space for developers to build applications that make use of the user's social circle. We can therefore experiment with social relationships by building social software and analyzing people's behaviour. People share web resources on Facebook, and especially to our interest they share music videos. This is clearly social recommendation through software.

Tagging Tags are free form labels assigned by users to resources. The process of tagging is in opposition with expert classification. This has been used successfully to organize web links, pictures, music, books and many more. The fact is that different people come up with different descriptors for resources and that the more resources are covered with all these descriptors the better.

Tag clouds With tagging appeared a new form of representing resource descriptors. A tag cloud displays a list of tags sorted alphabetically and with the size proportionate to the frequency of its usage. Tag clouds are used in many places on the web and give to the visitor a quick idea of what she will find on the portal, clicking on one tag of the cloud usually leads to a list of resources relevant to that tag.

Web APIs Many web portals offer access to their data through Web Application Programming Interfaces. This gives the researcher new resources to work from.

I believe that two important aspects of serendipity in human life are social encounters and language. These two aspects are intertwined as we learn most of our language by interacting with other people. My goal in this thesis is to show that using social network data in building navigation and recommendation tools fosters new discoveries of resources and terminology.

1.3 Concepts

I introduce two concepts to experiment with the use of social network data for music discovery, namely tag navigation and social shuffle.

1.3.1 Tag navigation

The principle of tag cloud navigation is to enable the exploration of large collections of resources by controlling a navigation tool called tag cloud. A tag cloud, as shown in Figure 1.1, displays a set of tags arranged alphabetically and shown with a size relative to their importance. When a user clicks on a tag from the tag cloud, the system adds the tag to the query, the list of selected keywords. The page is redisplayed and shows relevant items to the new query and a new tag cloud of related tags to continue the navigation. The list of selected keywords can be managed by the user by removing tags or adding new ones by using the tag cloud.

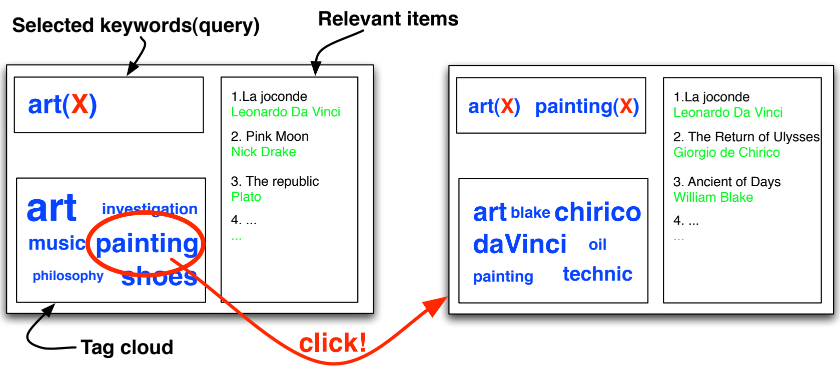

Figure 1.1. Principle of navigation with a tag cloud.

1.3.2 Social shuffle

People experience unexpected beneficial coincidences when listening music in a random order[45]. In [38] the authors show that social based recommendation outperforms standard information retrieval techniques by running simulations on users listening history, they perform their evaluation using precision and recall as metrics.

The idea of the social shuffle is to recommend tracks randomly and diffuse discoveries through the social network. A positive rating spawns diffusion through the user's social network, *i.e.* the shuffle recommendation becomes social.

Figure 1.2 gives an example of the process followed by the system when a recommendation is made by a user. In this case, Bob gets a random recommendation and gives it a 4 star rating, the video is then diffused to his social network. If a friend of Bob gives a high rating to the same video, it will be diffused to her network as well (in this case the example of Alice), if Bob's friend does not enjoy the video and gives it a 0 rating (the video is bad) then the social diffusion stops and here Chalie's friends will not get this recommendation.

1.4 Contributions

This thesis provides contributions in the fields of **Social media** and **Web engineering**, **Tag navigation**.

In the past few years the type of applications developed on the Web shifted and became more social with the appearance of social networks, blogs and more generally social media. Therefore the methodologies and tools to design and create digital applications have to be thought again. In earlier methodologies and technologies, social aspects were not taken into account. In this context I developed applications for Music discovery. In the current chapter I describe two concepts, Tag navigation, a mean to navigate collections of data using tag clouds and the Social shuffle, a mechanism to recommend and diffuse music within a social network. In Chapter 2 I give a litera-

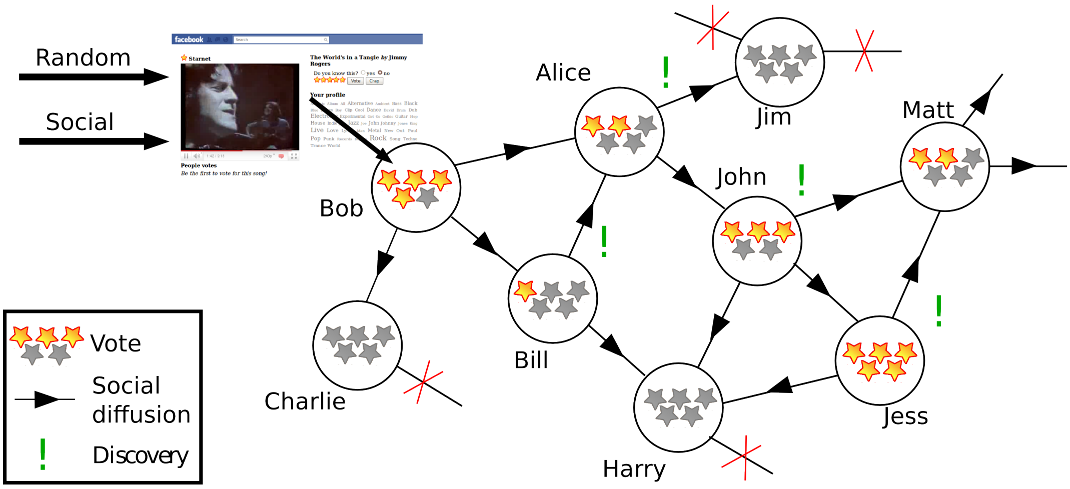

Figure 1.2. Diagram of the social shuffle principle. The chosen number of stars are yellow in color and lighter in black and white.

ture review of Music discovery and social tagging, showing that 1) social relationships are not currently used and studied in music recommendation systems 2) social tagging methods lack means for navigation.

I provided in Chapter 3 with three models to generate contextual tag clouds for Tag navigation, enabling someone to navigate a digital collection using a tag cloud. I built a data set from data gathered from the Last.fm Web API as described in Chapter 4. I compared these models in Chapter 5 within two web applications and three user experiments to test whether these models were practical to search for a particular song and to discover new music. The social and topic models each lead to more than 30% of music discoveries within the scope of the application developed.

I designed a social application with the Facebook and Youtube Web APIs for Music discovery and social interaction as described in Chapter 6 using a novel Web engineering methodology. I evaluated the discover functionality in Chapter 7 showing that music recommendations coming from someone's social network lead to more discoveries than the recommendations that are not social. 45% of social recommendations within the presented system are discoveries. The two distributions (social and non-social) are significantly different at the 0.005 confidence level.

To summarise my contributions here are the main points :

- A formal definition of bayesian models for tag navigation based on popularity, social relationships and topic models.
- A statistical comparison of the bayesian models in three live web-based experiments.

- A social application implemented in Facebook to recommend Youtube music videos based on random selection and social recommendation.

- An evaluation of the discover functionality of the social application by providing a statistical comparison of tracks ratings on social and non-social recommendations.

1.5 Dissemination

The work I have done during my research has led to several publications, listed here. Some are based on the work I did directly for the thesis and others ended up not being cited in thesis. All the papers are related to the work, some more than others. The bolded titles are directly related to the work presented in this thesis.

1.5.1 Journals

- A flexible integration framework for Semantic Web 2.0 applications. E. Oren, C. Mesnage, B. Heitmann, A. Haller, M. Hauswirth, and S. Decker, IEEE Software, Sep. 2007. [66]

1.5.2 Book chapters

- The Social Semantic Desktop - a new paradigm towards deploying the semantic Web on the desktop. A. Bernardi, S. Decker, L. van Elst, G. Grimnes, T. Groza, S. Handschuh, M. Jazayeri, C. Mesnage, K. Moller, G. Reif, M. Sintek (2008). In: J. Cardoso, M. Lytras. Semantic web engineering in the knowledge society. Hershey, Pennsylvania, USA, 290-313. ISBN 978-1-60566-112-4. [5]

- **Modern Web Application Development.** M. Jazayeri, C. Mesnage, and J. Rose. In: Andrea De Lucia, Filomena Ferrucci, Genny Tortora, Maurizio Tucci (eds.) Emerging Methods, Technologies and Process Management in Software Engineering. Wiley-IEEE Computer Society Press, February 2008. [35]

1.5.3 Conferences

- Extending Ruby on Rails for Semantic Web applications. C. Mesnage and E. Oren, In Proceedings of the International Conference on Web Engineering, Jul. 2007, Demo presentation.[61]

- The NEPOMUK Project – On the way to the Social Semantic Desktop. T. Groza, S. Handschuh, K. Moller, G. Grimnes, L. Sauermann, E. Minack, C. Mesnage, M. Jazayeri, G. Reif, R. Gudjonsdottir. In Proceedings of the Third International Conference on Semantic Technologies (I-SEMANTICS 2007), Graz, Austria, 2007.[24]

- Collaboration on the Social Semantic Desktop. G. Reif, T. Groza, S. Handschuh, C. Mesnage, M. Jazayeri and R. Gudjonsdottir, In UMICS 2007 (Ubiquitous Mobile Information and Collaboration Systems), Jun. 2007.[69]

1.5.4 Workshops

- **Music Discovery with Social Networks.** C. Mesnage, A. Rafiq, S. Dixon and R. Brixtel. The 2nd workshop on music recommendation and discovery. Oct. 2011, co-located with ACM-RECSYS.[62]
- **Serendipitous Social Shuffle.** C. Mesnage, R. Brixtel. The 1st workshop on music recommendation and discovery. Sep. 2010, co-located with ACM-RECSYS.[54]
- **Piloted Search and Recommendation with Social Tag Cloud-Based Navigation.** C. Mesnage and M. Carman. The 1st workshop on music recommendation and discovery. Sep. 2010, co-located with ACM-RECSYS.[55]
- **Tag navigation**, C. Mesnage and M. Carman. Proceedings of the 2nd international workshop on Social software engineering and applications, 29–32, co-located with ESEC/FSE 2009.[56]
- Social thinking to design social software: A course experience report. C. Mesnage and M. Jazayeri. Proceedings of the 2nd international workshop on Social software engineering and applications, 19–24, 2008.[59]
- **Specifying the Collaborative Tagging System.** C. Mesnage and M. Jazayeri, SAAW (2nd Semantic Annotation and Authoring Workshop) co-located with ISWC, November 2006.[58]
- Towards Global Collaborative Tagging. C. Mesnage and M. Jazayeri, Submitted to 'Mastering the Gap: From Information Extraction to Semantic Representation', April 2006.
- White Coats: Web-Visualization of Evolving Software in 3D. C. Mesnage and M. Lanza, VISSOFT : IEEE International Workshop on Visualizing Software for Understanding and Analysis, September 2005. [60]

1.6 Thesis outline

- Chapter 2 reviews the research literature in music discovery and social tagging accross various research fields, in human computer interaction, information retrieval and social dynamics;
- Chapter 3 defines the formal models of generation of tag clouds based on bayesian networks using popularity, and social relationships;

- Chapter 4 describes the live experiments and their implementation;
- Chapter 5 gives an analysis of the statistical results of the live experiments;
- Chapter 6 discusses the realization of the social network application.
- Chapter 7 analyses the data of a social network application of music discovery.

In Chapter 8 I give conclusions and report on open issues and future work.

Chapter 2

Literature Review

In this chapter I present, discuss and relate to my work papers and PhD theses relevant to the topic of this thesis. The first section is about music discovery and the second about social tagging.

2.1 Music discovery

Research interest in music discovery and the question of how people discover new music is quite recent. The best source of recent research in the domain is the first workshop on music recommendation and discovery held in September 2010. In this section we present papers presented in this workshop, in the Journal of New Music Research and the International Conference on Music Information Retrieval. I divided the work presented here into categories, namely: sociological analyses, recommender systems, exploring the long tail, music tagging, social networks and discovery tools.

2.1.1 Sociological analyses

Laplante in [43; 42] presents a prelimanary study of music discovery within a population of young adolescents. She reports on a social network analysis based on interviews of 6 participants who were asked to describe their behavior with music and to draw a representation of their social network map. This map organizes the people of the participants social network in various areas (school, relatives, neighbors and others) and uses concentric circles to indicate the strength of the participant's relationship with each person of her social network with whom they share music or discuss music. They are asked to draw a circle around those with whom they discuss music most often, to mark with an asterisk the persons whom they trust the most for music recommendations, and to draw a box around the name of those with whom they maintain a relationship essentially based on music. All participants affirmed that changes in their music taste reflects changes in their social network. The analysis revealed that music opinion leaders were perceived as good communicators, who are highly invested in music, and who are willing to share information with their friends. Three of the participants identified themselves as opinion leaders. Laplante then looks at music discovery in relation with the strength of the relationship between people who exchange music information. According to the theory of weak ties, weak ties lead more often to the exchange of new information. Laplante argues that this might not be the case for music, as she explains : "music preferences are considered too personal and subjective to trust recommendations from someone one does not know well". However, the exchange of music information strengthens relationships. The role of strong ties seems more important in music discovery, one participant discusses how he asks his best friend about his opinion on music he discovers. Most participants recognize their parents or one of their parents to be very influencial in terms of music discovery. In this thesis I look at the use of explicit social relationships to recommend music, in Chapter 7 I compare recommendations coming from a user's social circle and recommendations coming from outside of her network, showing that social recommendations lead to more music discoveries. This confirms the theory of Laplante that strong ties are more important for music discoveries.

In [81] Teller *et.al.* work on the question : "How do college students - who are both heavy consumers of music and of technology - go about finding new music in a digital age?". First, they may rely on their social networks for recommendations. Second, they may get ideas from exposure to mainstream media. Third, thanks to an ever-increasing slate of technological innovations, they may find suggestions through digital media. Their study consists of a paper-pencil questionnaire given in 2003-2005 to 328 students enrolled in sociology and communication courses on three different college campuses across the United States. They find that social network is still the main first method to discovering music, followed by traditional media and then information technologies. They conclude that technology will be used to reinforce existing social patterns and relationships, rather than transform them.

In [14], Cunningham *et.al.* look at how individuals purposefully or serendipitously encounter "new music". They performed a diary study in which 41 participants from a human-computer interaction course were asked to keep a diary during three days in which they noted each time they encountered new music the source and place this finding happened. They found in this population that most discoveries were made on the internet (21.8%) followed by radio (18.8%) and what could be understood as social, which in their study was categorized as conversations only accounts for 1.2% of the music findings. The remaining 41.8% are spreaded in order of importance between TV, CD, Public broadcast, Computer, Movie, Performance, Mp3 player and ringtone.

Baker in [3] asserts through quantitative survey and qualitative interviews that social networking could generate a strong future for the distribution of music. The study shows within MySpace that both artists and consumers react positively towards the distribution and discovery of new music through social media platforms.

2.1.2 Recommender systems

Music recommender systems are software systems which enable for the discovery of new music.

In [4], Barrington *et.al.* compare the Genius recommender system with a recommender based solely on acoustic similarity, one based on artist similarity and random. They evaluate the three recommenders by doing playlist generation, a user is presented with a seed song which he rates on a 5 point scale, then he is presented with two playlists generated from either of the four recommenders and can remove songs that do not fit into the playlists in relation to the seed song and rate the playlists on a 5 point scale. The user is then asked to compare the two playlists by stating which is better. They experimented with both showing the song names or not. They discover that seeing song and artist names has a significant effect on how a playlist is evaluated, indicating that recommender systems must be designed with applications in mind. They found that while Genius performs as well or better than the metadata and content-based systems on their test collection of popular music, it is unable to make recommendations from the large "long tail" of new, undiscovered music.

In [15] Fields *et.al.* study the social network of musicians in MySpace. They use complex network theory and audio content analysis. They show that the artist network topology is related to music genres by clustering the network into communities based on the topology and tags. They show that the artist social graph and the acoustic dissimilarity matrix encode different relations. They conclude that these two relations are different sources for music recommenders.

In [47], Levy *et.al.* describe a music retrieval system based on both social tags and audio content. Last.fm uses a combination of collaborative filtering and analysis of user-supplied tags for artists, albums and tracks. They analyse tag data from 5265 artists and show that a third have no tags for any of their tracks and another third have around 5 tags per tracks on average. This shows that tagging alone can not be used in a recommendation system, known as the cold start problem. They extract muswords (music descriptors, such as "female vocalist") for a track by identifying musical events within it, and then discretising timbral and rhythmic features for each region found. They combine tag data and muswords in a vector space and provide an analysis of results using different parameters and for various types of muswords.

In [75], Shardanand *et.al.* introduce social information filtering and describe its implementation in a system called Ringo which started to make personalized music recommendations in July 1994. Social information filtering makes use of users' ratings to recommend items to each other. They propose and evaluate four algorithms, namely: the mean square differences algorithm, the pearson r algorithm, the constrained pearson r algorithm, and the artist-artist algorithm. According to their results, the constrained pearson r algorithm which takes into account the positive and negative correlations performed best in terms of number of correct recommendations performed. The artist-artist and mean square algorithms performed better in terms of quality of the recommendations but provided with less recommendations. They also report on qualitative aspects of Ringo, mainly stating that feedback from users changed over time going from saying that the recommendations were poor to saying that the recommendations were amazing as the system was getting bigger and therefore had more data to work from.

Bogdanov *et.al.* [6] present three different approaches to content-based recommendation based on musical dimensions such as genre and culture, moods and instruments, and rhythm and tempo extracted from audio features. They compare with recommendations from Last.fm in a user evaluation with 11 users. They expect the proposed approaches to be suitable for music discovery in the long tail which has a lack of contextual information, and incorrect or incomplete metadata.

In [31], Herrera *et.al.* analyse temporal patterns in users listening history. They use playcounts from last.fm to find patterns in the selection of artists or genres for certain moments of the day or for certain days of the week. They show that for certain users what to play at the "right" moment is predictable and could be used in recommendation systems.

Tomasik *et.al.* [82] show that using linear regression performs better than using sum or max when combining multiple data sources for music information retrieval. They combine data from text mining web documents to extract tags, content-based audio analysis to find acoustic features, and collaborative filtering. They run their experiment on a set of 10 thousand songs and use Pandora dictionary as ground-truth.

Aman *et.al.* [1] give a review of explanations, visualizations and interactive elements of user interfaces in music recommendation systems. They present a taxonomy of dimensions, namely : transparency, scrutability, effectiveness, persuasiveness, efficiency and trust. They measure recommendation aids across multiple systems according to these dimensions, Pandora and Amazon are the systems with the most recommendation aids.

2.1.3 Exploring the long tail

The thesis of Oscar Celma[12] developped at the Music Technology Group in Barcelona is closely related to my work. Celma's thesis, entitled "Music recommendation and discovery in the long tail" focuses on the problem of recommending music in the long tail. It has four main contributions:

- a novel user-agnostic evaluation method (or network-based evaluation) for recommender systems, based on the analysis of the item similarity network, and the item popularity.
- a user-centric evaluation based on the immediate feedback of the provided recommendations.
- a system prototype, named *FOAFing the music*, to provide music recommendations based on the user preferences and her listening habits.

- a music search engine named *Searchsounds*, that allows users to discover unknown music mentioned on music-related blogs. *Searchsounds* provides keyword based search, as well as the exploration of similar songs using audio similarity.

The first contribution is different from my work as it is an evaluation based on the item similarity network and item popularity whereas all my experiments are user-centric, a system-centric evaluation prevents from analysing discovery as we do not know if the recommended song is known or not. The second experiment of Celma is closer to my work as it is user-centric, the evaluation gathered 288 subjects and compared three different recommendation systems, *i.e.* a collaborative filtering approach using last.fm, a content-based approach based on item similarity and a hybrid approach. The survey prompts the user with demographic questions and questions regarding the user's music knowledge, then for each recommended song the user says whether she likes the song or not and whether the song is known or not. This evaluation method is similar to the two experiments I conducted on music discovery, the first I conducted involved tag clouds and the goal was to evaluate which tag cloud generation method lead to more discoveries, the second experiment compared three recommendation methods which are different from the ones Celma tested, I compare namely a popularity based recommender, a social based recommender and a random recommender.

The systems developed are semantic systems, *FOAFing the music* makes use of the semantic web and item similarity whereas the system I developed *Starnet* makes use of people' social network and random recommendations. *Searchsounds* is a search engine and I was not interested in producing a search engine.

Fundamentally our approaches to music discovery differ in essence. Celma when speaking of shuffle and random playlists says:

> we believe that serendipity can be achieved by creating more personalised and clever playlists.

He uses semantic relations in music data whereas I use randomness on different sources, a popular source, a social source and random source.

In [46], Levy *et.al.* present their work in analysing the Last.fm recommendations to investigate the claims that recommender systems suffer from a popularity bias. They show that there is no evidence that recommendations and radio cause a systematic bias towards more popular artists. They built nevertheless a prototype recommender for long tail artists using item-based collaborative filtering with both scrobbles (Last.fm name for users listening history events) and tags.

Lee[44] *et.al.* propose a collaborative filtering recommendation algorithm which removes the popularity bias. In fact recommendations from collaborative filtering often lead to "obvious" recommendations as most popular songs are the ones recommended. Their solution is to recommend songs coming from the long tail of "expert" users and novel to the user. They evaluate their algorithm by producing a page of recommendations for Last.fm users and contacted them by private message. After seeing the page, the user is asked to rate the list of recommendations on how much they liked them and how much novel they were. The survey was completed only by 11 users and show that the recommended items were mostly novel and relevant.

In [19], Gaffney *et.al.* study the use of folksonomies for music discovery by users of social networking sites as a mean to discover items from the long tail. They examined in this project are MySpace, Lastfm, Pandora and Allmusic. They conducted interviews and questionnaires by contacting people outside concerts, directly on social networking sites and independent record companies. Although participants use social networking sites for music discovery, they found that people are still not using tags for discovery very much and that the ones who tag do it for personal future retrieval.

Research to unveil the long tail is still mainly oriented towards a use of collaborative filtering tweaked to explore the long tail. My thesis is to make use of social relationships to create new connections between people and music.

2.1.4 Music tagging

Lamere[40] reviews social tagging for music information retrieval. The paper describes the use of social tags on music discovery web sites such as Last.fm. It lists issues of tagging such as the cold start problem and gives an analysis of tags on Last.fm, out of 280 thousand artists only 21 thousand have at least one tag, therefore tagging has the problem of leaving a huge amount of data not indexed. Solutions proposed by research are tagging games and autotagging. Other problems are synonymy, polysemy and noise where people tag with different terms for the same concepts or mispelling. Hacking is another issue where malicious people tag with the purpose of controlling the system behavior, such as for instance tagging a new band with popular tags to increase the popularity of the band. Tagging is bias as most taggers are young people and the tag space reflects the interest of the taggers population and not the general population. Although tagging poses many problems, it is an interesting opportunity for music information retrieval research, namely regarding: expanding the tag coverage, using tags for discovery and improving the tag quality. What interests us in this thesis are questions regarding discovery, Lamere lists:

- How can we build an interface that exploits social tags to give a listener a more intuitive understanding of the interrelations between the many genres, styles and moods found in music?
- How can we use social tags to bridge the semantic gap, to allow listeners to find music by describing the music they like using words?
- How can we use social tags to give transparent explainable recommendations?
- How many social tags are enough before they can be used meaningfully for recommendation and discovery? Are 5 tags enough?, 10? 100?

Turnbull *et.al.* [85; 86] built a semantic music discovery engine based on both tags and audio content similarity. They designed a game for humans to tag songs and compare the data collected with data collected from surveys and internet music sites. To solve the cold start problem they developped an autotagging system based on audio analysis which is trained using the data collected in the tagging game. The resulting discovery engine prototype, named CAL, enables for query-by-description music search and radio playlist generation. Although it is claimed the system enables for discovery, there is no evaluation of the amount of music people discover using the system.

Fields[16] *et.al.* use topic models on tags to generate playlists. They use Latent Dirichlet Allocation as topic models which will be discussed in the next section of this chapter. From the extracted topics they generate playlists that are across genres but related. They do not provide with human evaluation of the generated playlists. Sordo[79] *et.al.* propose a mechanism to organize music tags with semantic facets. They use wikipedia terms structure to categorize tags from Last.fm.

Few research has been done on the interaction of people with music tags, most of the research focuses on how to help people to add tags or to tag music, not on how to interact with tag data. My interest is on interacting with tag data to search and discover information.

2.1.5 Social networks

Garg *et.al.* [20] examine how big a role social networks play in users' discovering new content by looking at peer influence in a dataset from Last.fm. They find that there is a positive influence of online peers on diffusion of new music. Their study is limited in size as they look at around 50 users and they use the Last.fm neighbors as peers. Neighbors are people with similar taste to the user. The problem with their study is that Last.fm neighbors reflect peers in a social network as these are people who do not know each other and are connected by the system itself. In this thesis I built an experiment using Facebook friendships to look at social diffusion.

2.1.6 Discovery tools

Good Vibrations[71] is a plugin for the Winamp program for music tagging and exploration. The tool enables the user to tag his music library and proposes recommendations based on the Semantic Web tool Foafing the music[12].

Lamere *et.al.* [41; 49] developed a tool called *Search Inside The Music* to visualize albums in 3D. The tool proposes various visualizations based on music similarity. Although the tool provide mechanisms to explore similar music, it is unclear how it enables for music discovery.

Magas *et.al.* present *mHashup*[51] a tool for finding similar music from the long tail. The tool finds similar tracks from audio chunks and present them for listening.

In [80] Stober *et.al.* present a zoomable interface for interactive exploration of music collections. By choosing weights for different aspects of similarity, the user can manipulate the projection and the neighborhood relations visualized through a lens. They do not provide with a user evaluation of their system.

Miller *et.al.* [65] present a tool for mobile devices, implemented for iPhones and Ipod touch, which enables for tag browsing and makes recommendations based on location. The application displays tags in a tag cloud where position of the tag is relative to the correlation between tags of similar songs. The location-aware feature extracts paths segments by looking at the user's GPS position change. The evaluation of the tag based feature consist of a user study of 14 participants. The evaluation of the location-aware recommendation is based on skipping behavior. They compare generating recommendations using random, song similarity, user's history and user's paths. They report the skipping behavior of one user who used the system for three weeks. The number of skips when using the paths is greatly reduced, indicating that location improves the quality of recommendations.

2.1.7 Conclusion

Music discovery is a recent field of research, it is interesting to note that most of the papers speak about serendipity which makes the field relevant for anyone who studies serendipity. My goal is to find mechanisms to experience serendipity, music discovery is a good place to start doing research on this topic. The questions around music discovery are deeply rooted in sociology as music is a social phenomenon in the way it is created, distributed and experienced. The discovery of items is either studied in recommender systems or with browsing and exploration with tagging. In the following section I give a literature review more specific about social tagging.

2.2 Social tagging

The first tagging applications sparkled in the late 90's with WebTagger[37]. Since 2004[28], social tagging reached a scale and momentum which made it more and more popular, mainly first for bookmarking tool with *del.icio.us* and then with the appearance of social media sharing sites such as flickr and youtube. Research in social tagging started slowly, Hammond[28] gives an overview of social bookrmarking tools, Golder *et.al.* [21] is the first analysis of tagging as a process and of tag data from del.icio.us. This paper lead to the first "Collaborative Web Tagging Workshop" co-located with WWW'06 [77], which papers mainly discussed tagging incentives, tagging applications (in museums, enterprises, tagging physical world), tag recommendation and knowledge extraction. Following this workshop, research in tagging has spread in various already established areas namely in web research, social dynamics, semantic web, information retrieval, human computer interaction and data mining.

Categories	Research papers
Social dynamics	[9][10]
Knowledge extraction	[72][26][25]
Community discovery	[2]
Generative models	[39][50][67]
Social search	[18][8]
Tagging incentives	[87][21][53][22][52][13]
Tagging applications	[37][33] [83][64][84][58]
Tag recommendation	[88][32][78]
Tag quality	[87][73]
Visual representations	[29] [70][30][27][36][76]
Tag navigation	[37][17][63][34][48]

Table 2.1. Categorization of research in social tagging.

2.2.1 Categorizing research in social tagging

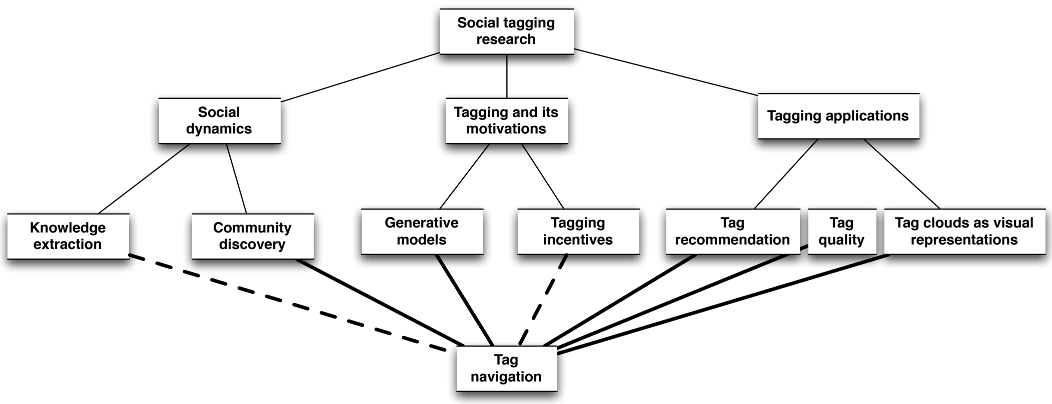

Figure 2.1. Taxonomy of research in social tagging.

In Figure 2.1 I give a taxonomy I made of the research which happened so far in social tagging. The main differentiation is between social dynamics, tagging incentives and tagging applications. Social dynamics is a field of physics which is concerned with understanding the behavior of groups of people from a statistical point of view, as opposed to sociology or anthropology. Tagging incentives research is looking at why people tag, what motivates them to tag and enumerating problems with tagging. Research in tagging applications deals with new applications of the principles of tagging in areas where it has not been applied yet.

I give here a short description of each of the categories used in Table 2.1. The table relates research papers in social tagging to the main category the research falls into.

Social dynamics The study of behaviors of social groups has evolved since the appearance of computing devices. Social dynamics models crowd behaviors, from a physics perspective, this can be the mechanics of the movement of crowds going out of a stadium in case of emergency with the goal to improve the positioning of exits. From an information perspective, the analysis of social data such as social networks brings understanding of how people connect to each other and how groups relate through information. Tag data is interesting for social dynamics as it shows group behavior and makes it possible to analyse in large scale the exchange of information between humans.

Knowledge extraction Knowledge is a mental construct of organisation between concepts. Taxonomies for instance are transcriptions of knowledge which can be used to conduct analyses or to refer to particular concepts and their relations. The construction of knowledge is a hard task which requires individuals to abstract and study topics in depth. Automatic knowledge extraction can be done through various methods, for instance automatic categorization of text documents leads to the making of taxonomies of the studied domain. Tag data enables researchers to extract common concepts and their relations from specific domains(*i.e.* music, photos, web pages, science).

Community discovery People sharing common interest or common aspects, such as location or company might be in the same community. The discovery of such community through the analysis of digital data like emails or structured data enables to put people in relation and analyze relevant information for the whole community. For instance if it is known that a particular person is part of a community interested in electronic music, the person might be interested in items bought by people in the same community. Tag data is a clearly a resource for community discovery.

Generative models Models which generate data are useful as tools for prediction. The training and validation of such models is made on empirical data, usually trained on a part of a dataset, we look if the model is able to predict the other part of the dataset. In tagging systems, generative models can be used for automatic tagging of resources, for tag recommendation or suggestion and for tag navigation to generate tag clouds.

Social search The emergence of social networks has lead to a new way to improve search of resources. The main idea is that people sharing common interest might look for similar resources. Social search uses the graph of social relations to rank resources according to a query. Tag data is a feature of most social networks and can be used as a means to describe users.

Tagging incentives Early research on social tagging was concerned with understanding why people tag. Understanding why people tag in successful systems enables engineers to reproduce such systems and to build appropriate solutions for each type of tagging incentive. This research lead to a categorization of tagging behaviors, such as tagging for future retrieval, tagging for advertising, tagging for personal management.

Tagging applications Tagging has already been successful in various applications, mainly multimedia sharing (pictures, video, music), bookmarking, blogs and scientific publications. How can we use the same principles for other domains? There has been some research on tagging in the enterprise, tagging in museums and tagging for scientists.

Tag recommendation Choosing a set of tags to assign to a resource is a hard task which requires effort by the user. To ease the process, systems give recommendation of tags already used by the user or by other people for the same resource. Depending on the tag recommendation mechanism used, different global behaviors might appear. If for instance popular tags are recommended, the system will converge faster as people would rapidly use the same tags. This might be desired by the system designers to reduce the size of the vocabulary. On the other hand, the entropy of a resource description diminishes if the tags used to describe it are the same used on most resources. Research in tag recommendation looks at solutions and analyzes the effects of the solutions.

Tag quality Tags are used as resource descriptors to present a resource in a list for instance a few tags would be displayed together with the resource title and other metadata. The quality of the tags displayed is very important as it is shown to every user. Some people might use tags with spelling mistakes, unreadable or vulgar. The notion of tag quality is to assess to quality of a tag to describe a particular resource. Research conducted in this field involves user studies voting within the system if a tag is relevant to the resource and games in which people have to agree on a tag to describe a resource.

Visual representations Tag data is often represented in tag clouds where terms are sorted alphabetically and the font size of the terms represent the importance of the term, usually the number of resources tagged with the term. Research in visual representations of tag data experiments with different ways of representing the graph tag data is composed of. Other attempts to represent tag clouds have been made, such as clustering the tags and position them based on their semantic similarity or representing tags in a graph showing the relations between them as edges.

Tag navigation Tags are used as a means to browse collections of resources. Anytime a tag is displayed, it is a hypertext link to a page listing resources tagged with this tag. When browsing a tag, systems often give related tags to the tag queried for further navigation. Tag clouds and related tags are navigation tools for tag data. Research in tag navigation looks at means to generate tag clouds in order to enhance the searchability of tagging systems when people are browsing tags.

2.2.2 Tag navigation

This section is about related work in tag navigation. Research in tag navigation is very sparse. For each paper described here I give limitations.

In [37], Keller *et.al.* present the first social bookmarking system in 1997. The system called Webtagger enables users to share bookmarks and assign tags to them. Users need to redirect their proxy through webtagger to install it, then buttons are displayed on the top of each page browsed, namely categorize, retrieve, view, comment and help. The approach is novel compared to storing bookmarks in the browser's folder in the sense that bookmarks are shared and they belong to multiple categories instead of being in just one folder. They argue that hierarchical browsing is tedious and frustrating when information is nested several layers deep. They do not provide with an evaluation of their system.

In [17], Fokker *et.al.* present a tool to navigate wikipedia using tag clouds. Their approach enables to select different views on the tag cloud, by recent tags, popular tags, personal tags or friends tags. They display related tags when the user moves the mouse over on a tag in the tag cloud. They do not give related tags to multiple tags queries.

In [63], Millen *et.al.* present the design of the *dogear* social bookmarking application. They analyse the log files to find empirical evidence that social tags improve social navigation. The application allows users to browse other people's bookmark collection by clicking on their username. Bookmarks collections can also be reached by clicking on a tag. The main page includes a popular tag cloud and individual user pages include individual tag clouds. They find that most browsing activity of the web site is done through exploring people's bookmarks and then tags. They show that by browsing tags people browse bookmark collections of other people.

In [48] Li *et.al.* propose various algorithm to browse social annotations in a more efficient way. They extract hierarchies from clusters and propose to browse social annotations in a hierarchical manner. They also propose a way to browse based on time. My opinion is that hierarchical browsing is not more efficient than tag cloud based navigation in the sense that hierarchical structures are not the same for everybody. Moreover they consider only the case of what they call sub tags or related tags to one tag at a time when I look at related tags to multiple tags.

The rest of the literature review presents work on top of which I build my research.

2.2.3 Visual representations

In [30], Hearst *et.al.* discuss the value of tag clouds. They convey the results of two qualitative studies. They conducted 20 interviews of people who are active in either web design or information visualization. They wondered in which way people thought the tag clouds as being useful, answers contained showing trends, seeing change of information, the availability of tags on the site, get the gist of the site, being playful, fun or inviting. Another use of tags is for self-reflection for people looking at their own tag clouds. Their second study is a web page analysis, they sampled pages returned by the Google search engine when searched for "tag clouds" usability, trends and navigation. They categorised the discussions into 20 categories. They quote the discussions to show different opinions on the negative aspects of tag clouds, the popularity or faddishness, the role of navigation, the impact on and reception by new users, trends and tag cloud data as social data.

Rivadeneira *et.al.* in [70] propose a paradigm for evaluating tag clouds and give guidelines for tag cloud construction. They identify tasks tag clouds can support, namely search, browsing, impression formation or gisting, and recognition/matching. They differentiate tag cloud features as text features and word placement. Text features concern the font weight, the font size and the font color, whereas word placement is affected by sorting, clustering and spatial layout. Their first experiment was conducted on thirteen participants, the goal was to examine the recall from visualizing a tag cloud. People were presented with tag clouds of thirteen words from psycholinguistic database positioned randomly and with different font sizes. People recalled better words with larger font size. The second experiment tested the effect of font size and word layout on impression formation, they displayed four types of tag clouds sorted as sequential - alphabetical, sequential - frequency, spatial and list by frequency. People again recalled better words with larger font size, the layout had no effect on recognition, there was a moderate effect of layout on impression formation where the tags displayed as a list ordered by frequency resulted in a better identification of the categories.

Halvey *et.al.* [27] conducted an experiment to evaluate the time taken to find a particular tag in lists represented with different layouts. They presented people with lists of 60 countries and the task was to find the one asked for. They found that horizontal alphabetical lists perform better at this task followed closely by vertical alphabetical lists and alphabetical tag clouds.

Sinclair *et.al.* in [76] conduct a study to examine the usefulness of tag clouds for information seeking. They asked participants to perform information seeking tasks on a folksonomy like dataset. They provided them with an interface consisting of a tag cloud and a search box. The folksonomy was created by the same participants who were asked to tag ten articles at the beginning of the study, leading to a small scale folksonomy much like the ones which could be found in small organizations or enterprises. The tag cloud displayed 70 terms in alphabetical order with varying font size proportional to the log of its frequency, they give the following equation :

$$TagSize = 1 + C\frac{log(f_i - f_{min} + 1)}{log(f_{max} - f_{min} + 1)} \tag{2.1}$$

C corresponds to the maximum font desired, f_i to the frequency of the tag to be displayed, f_{min} and f_{max} to the minimum and maximum frequencies of the displayed tags. Clicking on a tag of the tag cloud brings to a new page listing articles tagged with the clicked tag and a new tag cloud of co-occurring tags, clicking again on a tag restricts the list to the articles tagged with both tagged and so on. The search is based on a TF-IDF(term frequencyâĂŞinverse document frequency) ranking. Participants were asked 10 questions about the articles and then to tell if they preferred using the search box or the tag cloud and why. They found that the tag cloud performs better when people are asked general questions, for information-seeking, people preferred to use the search box. They conclude the tag cloud is better for browsing, enhancing serendipity. The participants commented that the search box enables for more specific queries.

In [29], Hassan-Montero *et.al.* propose an improvement of tag clouds by displaying them by similarity. They use the Jacard coefficient as measure of similarity, known as the relative co-occurrence. The relative co-occurrence is the division between the number of resources in which tags co-occur and the number of resources in which appear any one of two tags. If A and B are the resources tagged by two tags, the relative co-occurrence is :

$$RC(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.2}$$

They define a usefulness metric to select which tags to display in the tag cloud as the sum of the log of the frequency of a tag applied to a resource divided by the square of the number of tags assigned to the resource. The standard popularity metric being the sum of the frequency applied to resources for a tag. Their method provide little improvement on the coverage of the selected tags. The tag cloud layout is based on the similarity coefficient. The authors do not provide an evaluation of the tag cloud.

Kaser *et.al.* [36] propose a different algorithm for tag cloud drawing. Their methods concern how to produce the HTML in various situations. They also give an algorithm to display tags in nested tables. They do not provide evaluation regarding the usefulness of the new visual representations.

2.2.4 Tagging incentives

Golder *et.al.* [21] give a taxonomy of tagging incentives and look at convergence of tag descriptions of resources in *del.icio.us*.

In [74], Sen *et.al.* examine factors that influence the way people choose tags and the degree to which community members share a vocabulary. The three factors they focus on are personal tendency, community influence and the tag selection algorithm. They give five main dimensions for the tagging design space of a social tagging system, tag sharing where users are shown other people's tags, tag selection as the method a system uses to select tags to display on the screen, item ownership where people can tag other people's items, and tag scope as broad where a tag application is a (user, item, tag) triple or narrow where tag applications are (item, tag) tuples. Other dimensions concern constraints on the creation of tags, if a tag can contain spaces or special characters, what are tag delimiters. Their study focuses on the MovieLens system which consists of user reviews of movies. They categorize tags in three categories, factual tags, subjective tags and personal tags. They divided users of the system and assigned each group a different user interface, the unshared group would not see the community tags, the shared group saw tags from their groups using a random selection algorithm, the shared-pop displayed the most popular tags, and the shared-rec group used a recommendation algorithm. The recommendation algorithm selects tags that are most commonly applied to both the target movie and to the most similar movies. They find that habit and investment influence user's tag applications, that community influence affects a user's personal vocabulary. The shared group produced more subjective tags, some factual and a few personal; the shared-pop lead to more factual tags, a few subjective and personal; the shared-rec produced more factual tags some subjective and some personal. They also conducted a user survey in which they asked users to tell for which task they thought tagging was useful, self-expression(50%), organizing(44%), learning(23%), finding(27%), and decision support(21%).

[52; 53] Marlow *et.al.* propose a model of social tagging. Tags are associations between resources and users. They define a taxonomy of different aspects in the design of tagging systems that influence the content and usefulness of tags, namely tagging rights (who is allowed to tag?), tagging support (blind tagging, viewable tagging, suggestive tagging), aggregation (bag-model, set-model), type of object (e.g., web pages, images etc.), source of material (by participants, by the system, any web resource), resource connectivity (linked, grouped or none), social connectivity (linked, grouped or none). They also propose aspects of user incentives expressing the different motivations to tag, future retrieval, contribution and sharing, attract attention, play and competition, self presentation, opinion expression.

2.2.5 Social dynamics

[10; 9; 11] Cattuto et al. make an empirical study of some tag data from del.icio.us and find that the distribution of tags over time follows a power law distribution. More specifically they find that the frequency of tags obey a Zipf's law which are "characteristic of self-organized communication systems and is commonly observed in natural languages and written text". They reproduced the phenomenon by using a stochastic model, leading to a model of user behavior in collaborative tagging systems.

2.2.6 Tag quality

In [73], Sen *et.al.* question tag quality. Tagging systems must often select a subset of available tags to display to users due to limited screen space. Knowing the quality of tags helps in writing a tag selection algorithm. They conduct a study on the MovieLens system, this system collects movie reviews from users. They added in the interface a mechanism for users to rate the quality of tags. They experimented with multiple rating interfaces. All tags can not be rated, therefore they look for ways of predicting tag quality, based on aggregate user behavior, on a user's own ratings and on aggregate user's ratings. They find that tag selection methods that normalize by user, such as the numbers of users who applied a tag, performs better.

Von Ahn in [87] tackles the problem of tag quality by having people guessing tags used to index pictures, it gives a measurement to evaluate the tag quality for retrieval.

2.2.7 Generative models

In [32], Heymann *et.al.* define the social tag prediction problem. The purpose of social tag prediction is, given a set of tags applied to a set of objects by users, to predict whether or not a tag should be assigned to a particular object. Being able to predict applications of tags can lead to various enhancements, such as increase recall, inter-user agreement, tag disambiguation, bootstrapping and system suggestion. They collected tag data from the del.icio.us social bookmarking service and fetched the web pages for each saved bookmark. They analyse two methods, using page information and using solely tags. The first one is relevant in the situation of social bookmarking but does not apply in the case where the tagged objects are not web pages (*e.g.* images, songs, videos). They develop an entropy based metric which measures how much a tag is predictable. They extract association rules based on tag co-occurence and give measurements of their interest and confidence. They find that many tags do not contribute substantial additional information beyond page text, anchor text and surrounding hosts. Therefore these extra informations are good tag predictors. In the case of using only tags, predictability is related to generality in the sense that the more information is known about a tag (*i.e.* the more popular it is), the more predictable it is. They add that these measures could be used by system designers to improve system suggestion or tag browsing.

In [67], Ramage *et.al.* compare two methods to cluster web pages using tag data. Their goal is to see whether tagging data can be used to improve web document clustering. This work is based on the *clustering hypothesis* from information retrieval, "the associations between documents convey information about the relevance of documents to requests". The documents clusters are used to solve the problem of query ambiguity by including different clusters in search results.

In [23], Griffiths *et.al.* describe the latent Dirichlet allocation method to extract topics from a collection of documents. The problem is to discover the set of topics that are used in a collection of documents. They treat each topic as a probability distribution over words, viewing a document as a probabilistic mixture of these topics. The resulting classification is a soft classification, meaning that each word occurs in multiple topics with different probabilities. The computation is a equivalent to a Markov chain Monte Carlo which converges to the target distribution.

Considering there are T topics, the probability of the ith word in a given document can be written as :

$$P(w_i) = \sum_{j=1}^{T} P(w_i|z_i = j)P(z_i = j) \tag{2.3}$$

z_i is a latent variable indicating from which topic the ith word was drawn from, $P(w_i|z_i = j)$ is the probability of the word w_i under the jth topic and $P(z_i = j)$ the probability of choosing a word from topics j in the current document. The two main probability distributions are $P(w|z)$, which indicates which words are relevant to a topic and $P(z)$ indicates the importance of topics regarding a document. The computation involves two matrices θ and ϕ. θ represents T multinomial distributions over the W words where $P(w|z = j) = \theta_w^{(j)}$ and ϕ represents D multinomial distributions, where D is the number of documents, over the T topics such that for a document d, $P(z = j) = \phi_j^{(d)}$. The process is an expectation maximization of $P(w|\phi, \theta)$ using the previous equation. In this paper they actually use a much more efficient mechanism involving Gibbs sampling .

I give such a detailed explanation of this model as it will be used heavily in the following chapter on tag data.

2.2.8 Conclusion

Research on social tagging has bloomed recently accross multiple fields of research and from various aspects. There is still just few research specifically on tag navigation and these works do not tackle the problem of navigation based on a query list of tags, *i.e.* on multiple tags instead of just one tag as a context. I described areas related to tag navigation and gave an idea of the relevance for tagging. In the following chapters I will use some of the work presented here, especially the work on generative models and LDA.

2.3 Conclusion

In this chapter I discussed related work to music discovery and social tagging. The main conclusions of the literature review are that 1) social relationships are not currently used and studied in music recommendation systems 2) social tagging methods lack means for navigation. Therefore my research is targeted towards the usage and analysis of using social network data to discover music and to enhance navigation with tag clouds.

Chapter 3

Bayesian generation of tag clouds

In this chapter I describe the mathematical models I built in order to rank tags. Particularly I am interested in finding ways to exploit the social graph to compute tag similarity. I define tag similarity here as the probability that a tag is clicked by a user knowing that she has clicked a tag or set of tags before. One model is based on tag popularity, another on social network relationships and one is based on topic models. These models will be applied in the next chapter in my web experiments.

3.1 Preliminaries

This section defines the different basic operations which are practical when dealing with tag data.

W The set W represents the collection of tags known by a tagging system. We denote w a particular tag.

E The set E represents the collection of information elements index by a tagging system. We note e a particular information element.

P represents the set of people registered in a tagging system.

$tf(w,e)$ The relative term frequency of a tag w and an element e. This corresponds to the number of times the element e was tagged with the tag w by different people normalized by the sum of the counts. If $N_{w,e}$ is the count for a document then we have :

$$tf(w,e) = \frac{N_{w,e}}{\sum_{w' \in W} N_{w',e}} \tag{3.1}$$

$tf(w)$ The global term frequency of a tag w. This is equal to the sum of the relative term frequencies for that tag over all elements.

$el(w)$ The set of elements which were tagged with the tag w.

$tags(e)$ The set of tags applied to a particular element e.

Q The set of tags selected by the user as the query.

$$tf(w) = \sum_{e \in E} tf(w,e) \tag{3.2}$$

3.2 Generating tag clouds

Generating a tag cloud given a query is equivalent to rank tags based on the probability that these tags appear together with the tags selected in the query. I give here the derivation of $p(w|Q)$ by applying Bayes' rule.

$$p(w|Q) = \frac{p(Q|w)p(w)}{p(Q)} \tag{3.3}$$

for ranking we can drop the normalization by $p(Q)$ as it is the same for each tag w, which gives us:

$$score(w|Q) = p(Q|w)p(w) \tag{3.4}$$

I apply the naive Bayes assumption as we can consider the features as independent, in fact $p(w_1|w)$ and $p(w_2|w)$ are independent. We can derive $p(Q|w)$ into the product of its features :

$$score(w|Q) = p(Q|w)p(w) \leq p(w) \prod_{w' \in Q} p(w'|w) \tag{3.5}$$

The product can be approximated to the sum of the logarithm. The approximation keeps the ranking. We can compute the score for a particular tag as follows :

$$score(w|Q) \approx_{ranking} \log p(w) + \sum_{w' \in Q} \log p(w'|w) \tag{3.6}$$

Computing $p(w)$ is straightforward, we can use the global term frequency of the tag w. The probability $p(w'|w)$ is the probability of a tag w' to appear on a tag cloud given that the user clicked on the tag w, this probability is a measure of the similarity between the tags w and w'. In the following sections I give different means to compute these values based on the popularity, tag co-occurrence and the social network.

3.3 Tag similarity

A tag cloud can be composed of up to 100 tags. It is usually displayed on the side of the screen as an extra navigation facility. We distinguish here two mechanisms. The first time a tag cloud is displayed on the home screen, nothing is known about the navigation. On the other hand, once the user has clicked on a tag, we can change the display of the tag cloud according to what is known, this type of tag cloud is called contextual tag cloud as it depends on the current navigation context.

3.3.1 By popularity

Ranking tags by popularity on the home page give to users a global access point to the most prolific sections of the portal. The most popular tags are reachable from the popular tag cloud and displayed with a font size proportional to the amount of activity on that tag. A measure of the popularity of a tag is given in the following equation :

$$popularity(w) = \sum_{e \in E} \frac{tf(w,e)}{\sum_{w' \in W} \sum_{e' \in E} tf(w', e')} \tag{3.7}$$

We define the popularity of a tag w as the sum of the relative term frequencies of the tag with every known element normalized by the sum of relative frequency for all tags and all elements. We use the same measure to compute the font size of the tag displayed. The font size is proportional to the log of the popularity of a tag :

$$font\_size(w) = a * \frac{\log(popularity(w)) - low}{high - low} + b \tag{3.8}$$

where a, b are constants representing the maximum and minimum font sizes desired by the designer and $high$ represents the log of the highest popularity value for the selected set of tags to be displayed and low the log of the lowest. As we know from previous studies that popularity tag distributions follow a power law, meaning that there are very few popular tags and a large amount of less popular tags, using the log of the popularity decreases the visual difference between highly popular and less popular tags.

3.3.2 Contextual popularity

Once the user clicks on a tag, a page with results is displayed. The tag cloud contained on this page is a contextual to the current query. It shows related tags. To find the related tags by popularity we introduce a

$$contextual\_popularity(w|w_i) = \sum_{e \in el(w_i)} \frac{tf(w,e)}{\sum_{w' \in W} \sum_{e' \in el(w_i)} tf(w', e')} \tag{3.9}$$

We define the contextual popularity of a tag w given that we known w_i was clicked as being the sum of the relative term frequency of w and each element which was tagged with w_i, normalized by the sum of the relative term frequency for each tag and each element.

We want the contextual tag cloud to be recursive, therefore we need a method of computing the popularity of a tag given that more than one tag was clicked before, in the following equation Q represents the set of tags which were clicked by the user.

$$contextual\_popularity(w|Q) = \sum_{w_i \in Q} \sum_{e \in el(w_i)} \frac{tf(w,e)}{\sum_{w' \in W} \sum_{e' \in el(w_i)} tf(w',e')} \tag{3.10}$$

We use the same formula to compute the font size for the popular contextual tag cloud as we used for the popular tag cloud. The font size is then proportional to the log of the contextual popularity, the font size represents the amount of activity of a tag within the current context.

3.4 Ranking tags based on topic models

The problem with popular tag clouds is that it only displays tags from the top of the power law and does not give access to tags from the long tail. We introduce an abstraction, topic models to dig the long tail. Topic models are probability distributions used in information science to extract topics from collection of documents.

3.4.1 Tags topic models

In this section we describe the different representations used to deal with topic distributions.

Z The set Z represents the topics extracted by LDA.

θ is a matrix which stores the tag distributions over the topics, $\theta_{w,z}$ is the value of the probability of the tag w to belong to topic z.

ϕ is a matrix of distributions over the documents. $\phi_{d,z}$ is the value of the probability of d to belong in topic z.

3.4.2 The entry topic model tag cloud

When the popular tag cloud is displayed on a portal home page using popularity to rank their tags, the tag cloud displayed when using topic models displays on tags based on their abstracted importance. The following probability $p(w)$ sums the relative importance of a tag regarding each topic times the importance of the topic.

$$p(w) = \sum_{z \in Z} p(w|z)p(z) \tag{3.11}$$

In order to bias the tag cloud in such a way as increasing the "effect of infrequent topics" we can add a parameter δ in the range $[0,1]$ as follows:

$$score(w,\delta) = \sum_{z \in Z} p(w|z)p(z)^{\delta} \tag{3.12}$$

If delta equals 1, then the resulting tag cloud is equivalent to the popular tag cloud. The delta parameter enables us to control the effect of the topics.

3.4.3 Contextual topic model tag cloud

Once tag navigation starts and we know which tag has been clicked previously we can compute $p(w|w_i)$ which represents the probability of w to be clicked knowing that w_i has been clicked. The conditional probability of w given w_i had been clicked is the sum over all topics of the probability of w to belong to a topic weighted by the probability of that topic to be the one selected when w_i was clicked. It can be written as follows:

$$p(w|w_i) = \sum_{z \in Z} p(w|z)p(z|w_i) \tag{3.13}$$

where $p(w|z)$ is the probability of the tag w to belong to topic z, this probability is given by the θ matrix resulting of the LDA process. To compute $p(z|w_i)$ we apply Bayes' rule :

$$p(z|w_i) = \frac{p(w_i|z)p(z)}{p(w_i)} \tag{3.14}$$

for continuation, we formulate the probability of a tag to be displayed on the screen, given a set of tags which were clicked before Q.

$$p(w|Q) = \sum_{z \in Z} p(w|z)p(z|Q) \tag{3.15}$$

We use the topics models as an abstraction again which we resolve by summing over all topics. The prior becomes $p(z|Q)$, the probability of z to be selected by the set of tags Q which were clicked. We can compute it as follows, using Bayes' rule :

$$p(z|Q) = \frac{p(Q|z)p(z)}{p(Q)} \approx \frac{p(z)}{p(Q)} \prod_{w' \in Q} p(w'|z) \tag{3.16}$$

This is computationally intensive, we can use as an approximation the product of $p(w'|z)$ for all tags of Q. The product over all tags of Q is equivalent when used for ranking to the sum :

$$p(w|Q) \approx \prod_{w' \in Q} p(w|w') \equiv^{rank} \sum_{w' \in Q} \log p(w|w') \tag{3.17}$$

3.5 Ranking tags based on the social network structure

The social network structure gives us another means to rank tags for display. Knowing what other people are interested in based on their tag distributions can be seen as a recommendation system in the typical use "people who liked the item you are browsing/purchasing also browsed/purchased x". In this work we use this principle to rank tags to be displayed on the screen, the intuition is then "people who use the tag w you just clicked also use tag w'".

$u(w)$ is the set of users who have w in their vocabulary.

$f(u)$ is the set of friends of the user u.

The question of ranking w given w' is given by computing $p(w|w')$ which we can formulate simply as the number of users who use both w and w' divided by the number of users who use w'.

$$p(w|w') = \frac{|u(w) \cap u(w')|}{|u(w')|} \tag{3.18}$$

This definition is based on a binary view of tag usage, either someone used a tag or didn't use it. A more precise definition based on the usage frequency of tags by users can be given as :

$$p(w|w') = \frac{1}{|u(w')|} \sum_{u \in u(w')} p(w|u) \tag{3.19}$$

where $p(w|w')$ is defined as the sum of the probability of a user to use a tag w over all users who used w' normalized by the number of users who use w'. There are multiple ways to estimate $p(w|u)$. We speak of an estimate as we consider that we have only a partial view of a user's interest.

The maximum likelihood estimate of $p(w|u)$ is straightforward, it is the term frequency of the usage u made of w divided by the sum of the term frequency that u made of the whole vocabulary.

$$\hat{p}_{ML}(w|u) = \frac{tf(w,u)}{\sum_{w \in V} tf(w,u)} \tag{3.20}$$

Based on the idea that we have not seen enough data to have a good measure of $p(w|u)$, we can compute a linear interpolation by weighting with a λ the term frequency of usage u made of w and by $(1-\lambda)$ the usage every user made of w.

$$\hat{p}_{LI}(w|u) = \lambda \frac{tf(w,u)}{\sum_{w \in V} tf(w,u)} + (1-\lambda) \frac{tf(w)}{\sum_{w \in V} tf(w)} \tag{3.21}$$

The Dirichlet smoothing estimate is given as follows :

$$\hat{p}_D(w|u) = \frac{tf(w,u) + \alpha \frac{tf(w)}{\sum_{w' \in V} tf(w')}}{\sum_{w \in V} tf(w,u) + \alpha} \tag{3.22}$$

3.5.1 The words of my friends

Another way of estimating the probability of a user u to use a tag w is by doing a linear interpolation of the relative frequency of the data we have seen and the sum of the frequencies of how much the friends of u use w.

$$\hat{p}_F(w|u) = \lambda p(w|u) + \frac{1-\lambda}{f(u)} \sum_{u' \in f(u)} p(w|u') \tag{3.23}$$

This gives us a way to compute the probability of a tag w to be clicked given that w' has been clicked before based on the social network of the people who use w'.

$$p(w|w') = \frac{1}{|u(w')|} \sum_{u \in u(w')} \frac{1}{|f(u)|} \sum_{u' \in f(u)} p(w|u') \tag{3.24}$$

It is defined as the sum over all users who used w of sum of the probability of their friends to use w, normalized by the number of friends of u and the number of users who use w.

3.6 Ranking tags for tag descriptions

A tag description is a short list of tags displayed to describe a particular information element within a list of results. The ranking used typically is by popularity which ranks the tags according to their relative term frequency. The ranking function can be written as follows where w is the tag to rank and e the information element which is displayed.

$$rank\_by\_popularity(w,e) = tf(w,e) \tag{3.25}$$

The problem with using the popularity is that the same tags get to be displayed for each of the elements of the results list. Following the power law distribution of the tags, a few tags are very popular and the list being a list of related items (they were the results of one same query), it is more likely that the same tags will be displayed for each element. To enhance the diversity of the tags displayed along the list we use a measure of dissimilarity between the tags distribution of the element and the tags distribution of the result list.

$$rank\_by\_dissimilarity(w,e,S) = p(w|e)log\frac{p(w|e)}{p(w|S)} \tag{3.26}$$

$p(w|e)$ is the probability that a tag is applied to the element e. It is the relative term frequency of the tag for the element divided by the sum of the relative frequencies for each other tag and that same element.

$$p(w|e) = \frac{tf(w,e)}{\sum_{w' \in W} tf(w',e)} \tag{3.27}$$

$p(w|S)$ is the probability of a tag w given the collection S of results. It is the sum of the relative frequencies for the tag and all elements of S divided by the sum of the term frequencies for all tags and all elements of the result set. It is equivalent to the sum of the probabilities for each element of the set

$$p(w|S) = \frac{\sum_{e \in S} tf(w,e)}{\sum_{w' \in W} \sum_{e' \in S} tf(w',e')} = \sum_{e \in S} p(w|e) \tag{3.28}$$

3.7 Conclusion

In this chapter we have defined the mathematical models of different ways of computing a ranking of the tags of the vocabulary of a system to display them on the screen. The goal being to display the tags that are most likely to be clicked. The contextual ranking is the ranking of the tags based on what tags have been clicked before by the current user. We give the formal definitions based on the popularity of the tags, on topic models and on the social network structure.

In the next chapter we look at how to apply these models in a running web application. The implementation consists of a Ruby on Rails application and database pre-computations.

Chapter 4

Piloted Search and Recommendation with Social Tag Cloud-Based Navigation

In this chapter I detail models for generating context-aware tag clouds using both social network and topic modeling based approaches, that I have implemented in our prototype tag cloud-based navigation system. I then describe the data we have collected from the Last.fm online music social network, and the evaluation consisting of a pilot user-study, a user survey and a follow up study.

Contribution This Chapter is a contribution to **Tag Navigation**. Navigation using tags has been developped recently in web based systems. In this chapter I look at tag navigation using multiple tags and using either social network data, popularity and extracted topics which is a new contribution.

4.1 Tag Cloud based Navigation

In this section we describe algorithms for generating context-aware tag clouds and query results list for tag cloud based navigation. Generating a tag cloud simply involves selecting the one hundred tags which are the most probable (to be clicked on by the user) given the current context (query). Estimating which terms are most probable depends on the model used as we discuss below.

Figure 4.1 shows the screen of the Web application we developed to evaluate our models. The goal is to find the displayed track using the tag cloud. The tag cloud is generated according to a randomly selected model and the current query. Participants in the evaluation can add terms to the query by clicking on tags which generates a new tag cloud and changes the list of results. Once the track is found, the user clicks on its title and goes to the next task.

Figure 4.1. Searching task.

Figure 4.2 shows the principle of our controlled recommendation experiment. The participant sees a tag cloud, by clicking a tag she is recommended with a song. Once the song is rated, a new tag cloud is given according to the previously selected tags.

4.1.1 Generating Context Aware Tag Clouds

We now investigate three different models for generating context-aware tag clouds. For each model we describe first how an initial context-independent cloud is generated. We then describe how the context dependent cloud is generated in such a way as to take the current query (context) tags into account.

Popularity based Cloud Generation Model

The first and simplest tag cloud generation model is based on the popularity of the tags across all documents in the corpus. We first describe a query independent tag cloud, which can be used as the initial cloud for popularity based navigation.

Ranking tags by popularity on the home page gives users a global access point to the most prolific sections of the portal. The most popular tags are reachable from the popular tag cloud and displayed with a font size proportional to the amount of activity on that tag. A measure of the popularity of a tag across the corpus is given in the following:

$$p(w) = \frac{\sum_{d\in D} N_{w,d}}{\sum_{d\in D} N_d} \tag{4.1}$$

Figure 4.2. Controlled recommendation task.

where $N_{w,d}$ is the count of occurrences of tag w for resource (document) d and $N_d = \sum_{w \in V} N_{w,d}$ is the total count for the document.

We can now compute a context sensitive version of the popular tag cloud quite simply as follows:

$$p(w|Q) = \frac{\sum_{d \in D(Q)} N_{w,d}}{\sum_{d \in D(Q)} N_d} \tag{4.2}$$

Where $D(Q) = \cup_{w \in Q} D(w)$ is the union of all resources that have been tagged with words from the query Q.

Social Network Structure based Cloud Generation Model

We are interested in taking advantage of additional information contained in the social network of users (friendships) in order to improve the quality of the tag cloud. We assume that the friends of a user are likely to share similar interests and thus we can use the tag description of a user's friends to smooth the tag description of the user.

We calculate an entry (context independent) social tag cloud as follows:

$$p(w) = \sum_{u \in U} \sum_{u' \in f(u)} \frac{N_{w,u'}}{\sum_{w \in W} N_{w,u'}} \tag{4.3}$$

where $f(u)$ is the set of friends of user u and U denotes the set of all users in the social network.

We apply a slightly different derivation to calculate the context dependent social tag cloud. We estimate the probability $p(w|w')$ given the context tag w'. These probabilities are precomputed and combined depending on the query at run time. We hypothesize that users who are friends on a social tagging website are likely to have similar interests (likes & dislikes) and that we can use the social network structure to improve contextual tag cloud generation. We can leverage the social network (by marginalizing out the user u) as follows:

$$p(w|w') = \sum_{u \in U} p(w, u|w') \tag{4.4}$$

$$= \sum_{u \in U} p(w|u) \frac{p(w'|u)p(u)}{p(w')} \tag{4.5}$$

Calculating $p(w')$ and $p(u) = N_u / \sum_{u' \in U} N_{u'}$ is straightforward. We compute $p(w|u)$ by summing over tag counts $N_{w,u'}$ for users in the social network of the user u:

$$p(w|u) = \frac{\sum_{u' \in f(u)} N_{w,u'}}{\sum_{u' \in f(u)} N_{u'}} \tag{4.6}$$

Note that since the summation in Equation 4.5 over all users involves a very large computation, we perform the summation only over the top 200 users as ranked according to the frequency $p(w|u)$.

Topic Model based Cloud Generation Model

Another way to smooth the relative term frequency estimates and thereby improve the quality of the tag clouds generated is to rely on latent topic modeling techniques [23]. Using these techniques we can extract semantic topics representing user tagging behavior (aka user interests) from a matrix of relationships between tags and people. Topic models are term probability distributions over documents (in this case users) that are often used to represent text corpora. We apply a commonly used topic modeling technique called latent Dirichlet allocation (LDA) [23] to extract 100 topics by considering people as documents (and tags as their content).

The entry (context independent) tag cloud based on topic modeling is defined as follows:

$$p(w) = \sum_{z \in Z} p(w|z)p(z) \tag{4.7}$$

Where $p(w|z)$ denotes the probability of the tag w to belong to (being generated by) topic z, its value is given as an output of the LDA algorithm. $p(z)$ is the relative frequency of the topic z across all users in the corpus.

To compute the context aware tag cloud based on topic modeling, we simply marginalize over topics (instead of users):

$$p(w|w') = \sum_{z \in Z} p(w|z)p(z|w') \tag{4.8}$$

$$= \sum_{z \in Z} \frac{p(w|z)p(w'|z)p(z)}{p(w')} \tag{4.9}$$

4.1.2 Ranking Resources

We follow a standard Language Modeling [7] approach to ranking resources (documents) according to a query. Thus we rank resources according to the likelihood that they would be generated by the query, namely the probability $p(d|Q)$, where d is a resource and Q the query as a set of tags. We give here the derivation of $p(d|Q)$ by applying Bayes' rule.

$$p(d|Q) = \frac{p(Q|d)p(d)}{p(Q)} \tag{4.10}$$

For ranking we can drop the normalization by $p(Q)$ as it is the same for each resource d, which gives us:

$$score(d|Q) = p(Q|d)p(d) \tag{4.11}$$

We apply the naive Bayes assumption and consider the words in the query to be independent given the document d. Thus $p(Q|d)$ factorizes into the product of word probabilities $p(w|d)$:

$$score(d|Q) = p(Q|d)p(d) \approx p(d) \prod_{w \in Q} p(w|d) \tag{4.12}$$

This product is equivalent in terms of ranking to the sum of the corresponding log probabilities. Thus we compute the score for a particular tag as follows :

$$score(d|Q) =_{ranking} \log p(d) + \sum_{w \in Q} \log p(w|d) \tag{4.13}$$

Computing $p(d)$ is straightforward, we can either use the length of the tag description of the resource d or the uniform distribution $p(d) = 1/D$ where D is the count of documents in the corpus.

For the browsing experiment, the log probabilities within the summation are exponentially weighted so as to give preference to the most recently clicked tags, as follows:

$$browsing\_score(d|Q) = \log p(d) + \sum_{i=1}^{|Q|} \alpha^{i-1} \log p(w_i|d) \tag{4.14}$$

Here w_i denotes the $i^{\mathtt{th}}$ most recent term in the query Q, and α is a decay parameter set to 0.8 in our experiments.

4.1.3 Precomputation

For each model we precompute the values for $p(w|w')$ which gives us three matrices of relations between tags. At run time we rank the tags to generate a contextual tag cloud according to a query of multiple tags as follows:

$$p(w|Q) = \beta \log p(w) + \sum_{w' \in Q} \log p(w|w') \tag{4.15}$$

In our experiments we set the parameter β to 0.5.

4.2 Empirical setup

We choose Last.fm to fetch our experimental dataset. Last.fm is a music sharing online social network which allows one to get social network data and tagging data from their application programming interface (API). To our knowledge it is the only network which enables researchers to fetch the friends of any user in the system. Fetching the social network is essential for experiments with social tag clouds.

We gather tag data by crawling users via their friend relationships. Once a new user is fetched, we download her own tags and then recursively fetch her friends and so on. We start by fetching the network of the author. In order to get a complete subset of the social network of Last.fm, we apply a breadth first search by exploring recursively the relations of each user. Once we have a substantial subset of the social network and tags, we fetch the tracks assigned to the tags. For each tag fetched, we get the 50 top tracks annotated with this tag.

Table	Size
People	126035
Friends	3523626
Tags	343681
Tracks	435257
Usages	900259
Tag applications	4236024

Table 4.1. Dataset size

Table 4.1 reports the size of the main tables of the database. The database accounts for more than 120 thousand people having 3.5 million friend relationships which makes

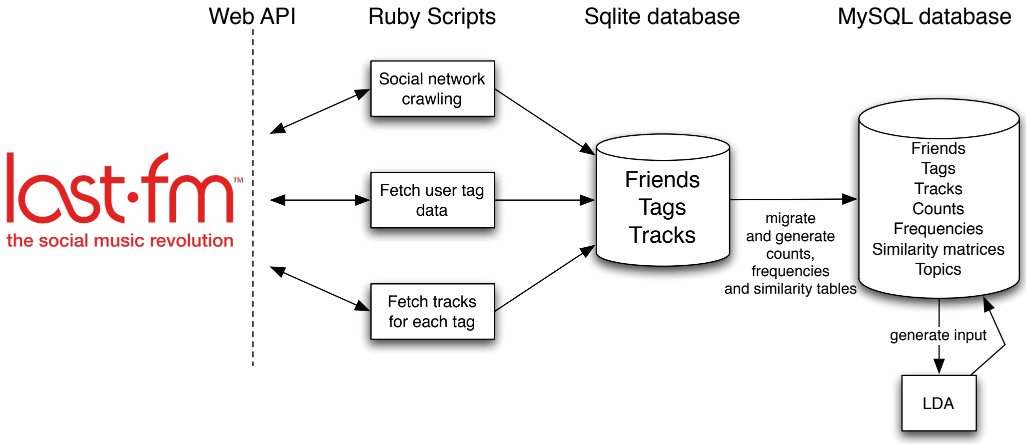

Figure 4.3. Fetching process

an average of 27 friends per person. These individuals have used more than 340 thousands unique tags a total of more than 4 million times, which makes an average usage of 12 times per tag. The total number of usages(what I call usage here is the fact that a person used a tag) is over 900 thousand which makes an average usage of 3 people for each tag.

Figure 4.4 shows the degree distribution of the number of friends. It shows the frequency of people with respect to the number of friends they have. The plot axes are the log of the values for better visualization. The plot shows a power law distribution in the number of friends per person with a number of friends superior to 10. Below ten friends, we have not seen enough data to have a good estimation of the distribution of the number of people with that many friends, so the distribution is curved. Power laws have been observed in other social networks and show that social networks are scale-free. Tag usage also shows a power law distribution.

Figure 4.5 is the log-log plot of the distribution of the tag usage. It shows the number of tags that have been used a certain number of times. The plot clearly shows a power law distribution.

Figure 4.6 shows the distribution of people with respect to the size of their vocabulary. The plot shows a power law distribution. Many people have a small vocabulary and people with a large vocabulary are often the only ones with such a vocabulary size.

Table 4.2 shows the most frequent terms of the dataset together with the number of occurrences and frequency given in percentage of the term. The tag *rock* accounts for 2.5% of the whole tag usage. This table corresponds to the top tags displayed on the entry tag cloud of the popular view. When compared with the popular tag cloud of last.fm, the list seems representative of the same proportions.

Once the data is fetched by the ruby scripts via the Last.fm Web API, we migrate it to a MySQL database for processing. We precompute various tables to store data that will

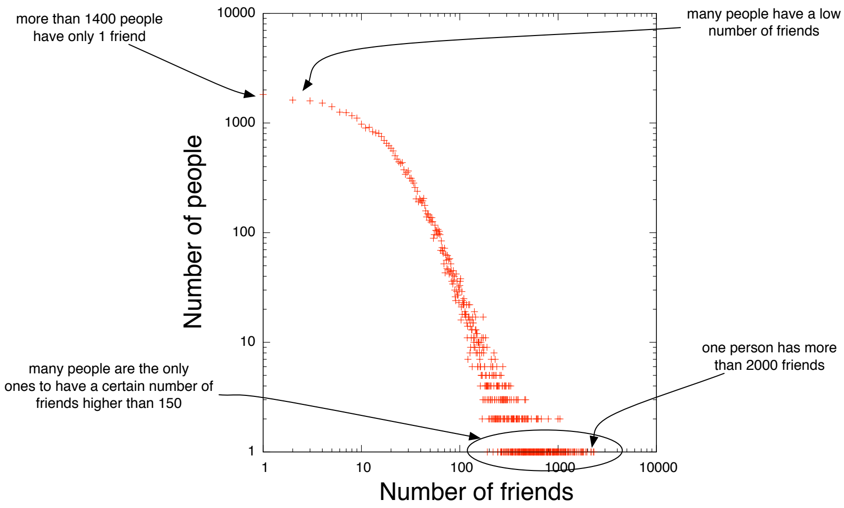

Figure 4.4. Plot of the distribution of friends.

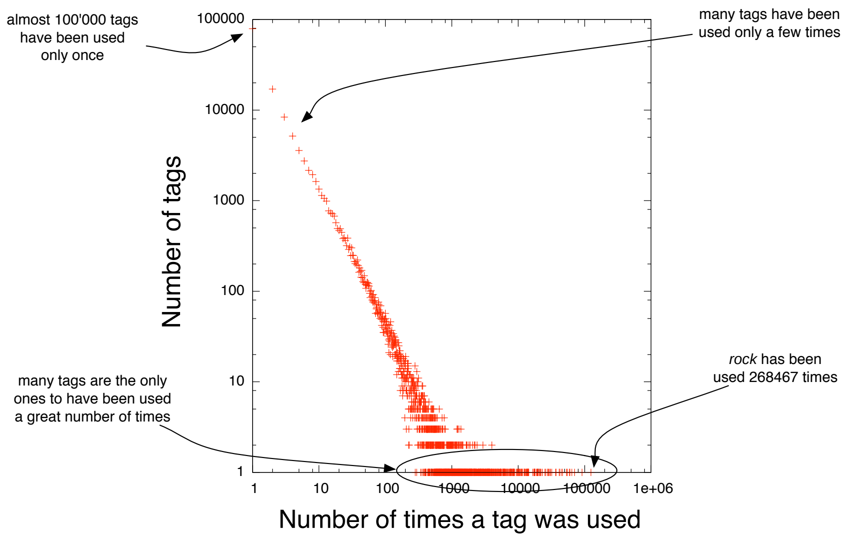

Figure 4.5. Plot of the distribution of tags.

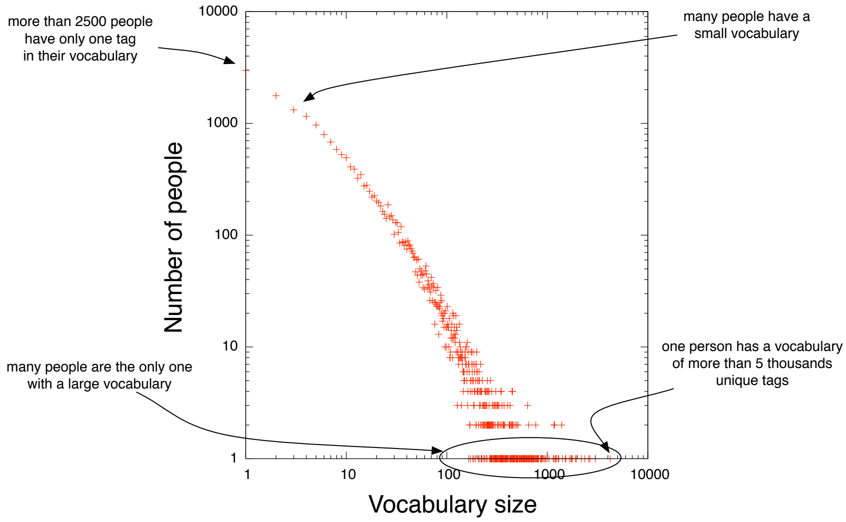

Figure 4.6. Plot of the distribution of vocabulary size.

be used multiple times in the calculations. For instance we compute the term frequency of each tag, the term frequency for each tag and each user, the frequency of the friends of a user for a tag. From these tables we can then compute similarity tables between the probability of one tag given another for each model which corresponds to $p(w|w')$, we do this only for the tags used by at least 5 people which accounts for about twenty thousand tags.

4.3 Conclusion

In this chapter I presented the set up for my experiments on tag navigation with the following contributions :

- Bayesian models for tag-cloud generation and their computation.
- A social dataset including music tracks, social tags, listeners and their social relationships.

In the following chapter I present the experiments made with these models and this dataset.

Frequent term	**Nb of occurrences**	**Percentage**
rock	268467	2.5198
electronic	193881	1.8197
seen live	167440	1.5716
indie	155891	1.4632
alternative	155200	1.4567
pop	125061	1.1738
jazz	115378	1.0829
female vocalists	109737	1.0299
ambient	88548	0.8311
classic rock	85290	0.8005
experimental	80976	0.7600
singer-songwriter	79607	0.7471
electronica	74642	0.7005
folk	68772	0.6454
metal	67331	0.6319
total	10654083	100

Table 4.2. Last.fm frequent terms ordered by number of occurrences.

Chapter 5

Tag Navigation Experimental Results

I built a web application to evaluate my models in a user study. I conducted a pilot study where tag clouds are used to search tracks, a user survey and a follow-up study with the search task and a browsing task where participants used the tag cloud to pilot a recommendation system. We find statistically significant evidence that the topic model and the social model perform better to generate tag clouds that lead to recommend songs that were liked and unknown by the participants than our base line, the popular model.

Contribution This Chapter is a contribution to **Tag Navigation**. I describe how to set up experiments to test the usefullness of tag navigation when searching or being recommended of an information item, more particularly for Music Discovery.

5.1 Experiment 1: Search with a tag cloud

5.1.1 Purpose

The purpose of this experiment is to evaluate the models presented in the previous chapter when used for tag navigation to search for a particular item.

5.2 Task

The study took place at the university of Lugano. We gathered 17 participants from our Bachelor, Master and Phd programs. Participants registered on an online form before the evaluation. They were asked to fill in an entry form and an exit form to answer general questions. The participants are asked to perform 20 tasks in which they must find a particular track as fast as possible. To find a track they click on a tag from the tag cloud which changes the list of tracks and displays a new tag cloud related to the tag selected, they can select another tag or remove the selected tag. Tracks are selected randomly from a pool of the 200 most popular tracks. The tag cloud generation method is also selected randomly for each task.

5.2.1 Method

The evaluation is designed as a within subject study. Each participant is her own control group as a model is randomly selected for each task and the participant is not directly informed of which model is used. Each action of the participants is stored in a log in the database.

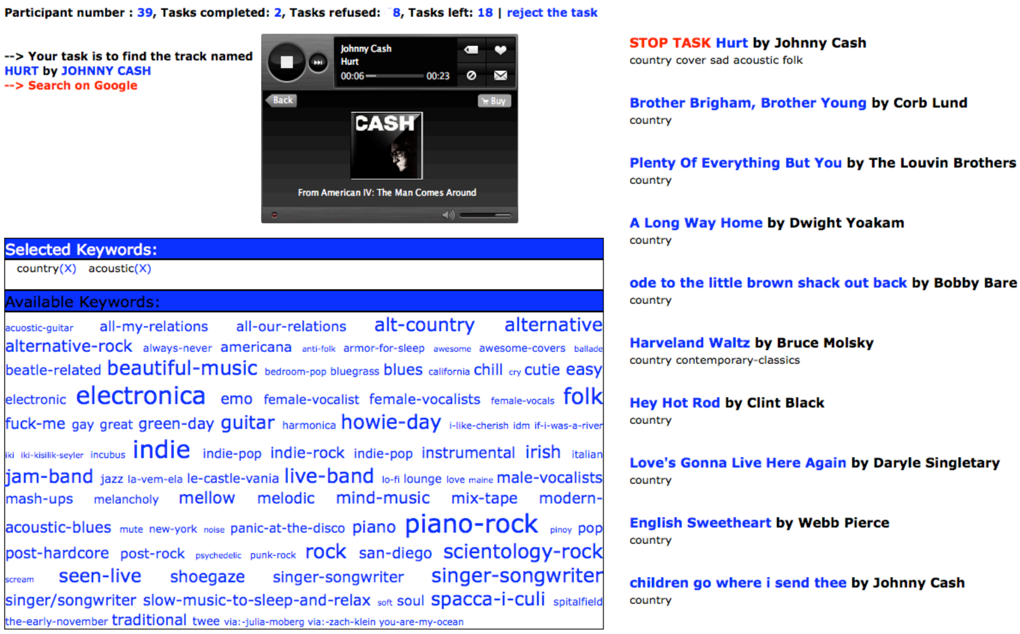

Figure 5.1. Screenshot of an evaluation task.

Figure 5.1 shows the popular tag cloud when queried for multiple tags, in this case the query consists of two tags: *country* and *acoustic*. The user can remove any tag from the query by clicking on the *X* link next to the tag. The associated tag cloud contains terms that are relevant to both tags in the query.

5.2.2 Results

Most participants had **fun** during the experiment. This is important and relevant as it keeps the participants motivated.

A participant noticed that quickly he was selecting popular tags and browsing through the track list for the "red link"[1] to stop the task. This technique had him finish with the

[1] the red link, named "STOP TASK" in the interface is displayed next to the track to find in the track list and enables to end the task and start a new one.

second place, we believe the first finishing participant had the same technique and was rejecting tasks[2] faster if he couldn't find it with popular tags. From the comments given, a participant gives as advantages "you don't have to think about the search terms, you can just pick one", another one adds "relief from typing". It seems to be the major advantage of tag navigation, it is hard for a person to come up with search terms from the vocabulary he has in mind, whereas when presented with a vocabulary, it is simple for him to choose what terms to use. Multiple participants think it would be simpler for them to type search keywords when they know beforehand what terms they would use rather than browsing the tag cloud to find the term they are looking for. Again it seems tag clouds are good to help remembering terms and when the participant does not know what terms to use, but in the case the participant has knowledge of what he is looking for it is easier for her to type. A participant notes "if a tag is not in the list, I can not use it. Free search would be better from this point of view".

Some participants mentioned as an advantage "discovering new music"[3]. Probably the evaluation process by itself makes the participant discover new music by selecting randomly a track from the 100 most popular tracks. Also people discovered new music by reading the list of tracks when they clicked on tags. A participant mentioned that he would like a tag cloud to navigate pages from his browsing history in his web browser. A tag cloud would help remembering topics he has seen in his browsing life.

Model	Started	Completed	Rel. Frequency (%)
Popular	132	94	71.2 ±3.9
Topic	131	93	71.0 ±4.0
Social	158	116	73.4 ±3.5

Table 5.1. Completed tasks per model. The rate of task completion along with the standard error in the estimate is given in the last column. The models are not found to be statistically significantly different from one another.

A total of 302 tasks were completed and 101 were rejected. Each time a new task is given the model used to generate the tag cloud is selected randomly from the three models available. 94 tasks were completed for the popular tag cloud and 94 as well for the tag cloud based on topic models. The tag cloud based on social network lead to 116 completed tasks. Table 5.1 summarises the number of started and completed tasks and gives the relative frequency in percentage for each model. The relative frequency

[2]Participants had the ability to reject a task by clicking the "REJECT TASK" button in case they had no clue how to find the track to search for, it would give them another track.

[3]The participants answered an exit survey in which question 13 was "I think that tag cloud navigation helps discovering new music". This is relevant as further on I decided to focus my research on Music Discovery. This is I think the serendipitous event as this was not what I was looking for in the first place, I was trying to revolutionize the typical text box search paradigm and on the way I found a lively and passionating research topic.

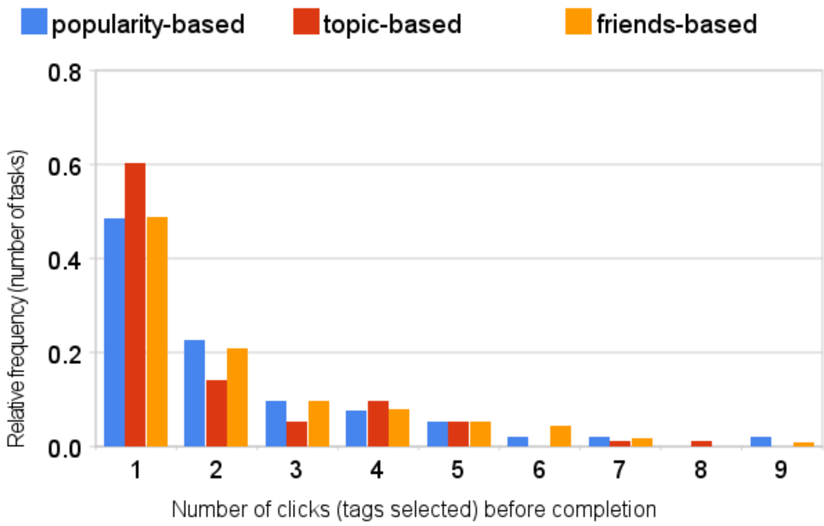

Figure 5.2. Histogram of different navigation path lengths across the three cloud generation models.

of completed tasks regarding the number of started tasks for each model is similar.

Figures 5.2 and 5.3 give an overview of the results. Figure 5.2 represents the relative frequency, the number of tasks completed with that number of tags clicked relative to the total number of clicks for each model. We see that most of the tasks were completed after the first click[4]. The tracks to find were selected from the top 100 popular tracks in our dataset. These tracks have a high probability of containing a popular tag.

We have graphed the data to show differences in the distribution of click-counts (navigation path lengths) and time to completion (time to find a song). On average, the time taken to complete a task is slightly shorter for topic-based tag clouds than the popular one (390 seconds against 400 seconds) and a bit better for the social based tag cloud (320 seconds against 400 seconds). While the distributions do vary slightly: the topic based model appears to have slightly lower navigation path lengths, and time to success values, the differences are minimal and the results are not considered conclusive nor statistically significant.

5.3 Follow-up study

We conducted a second study for which we adapted the system based on the comments we received in the pilot study. We improved the efficiency of the system by precomputing term relational matrices ($p(w|w')$). For this evaluation we had 20 participants. None

[4]For easy tracks there was no need to select multiple tags, one was enough, for instance clicking on "Rock" would display the most popular rock songs.

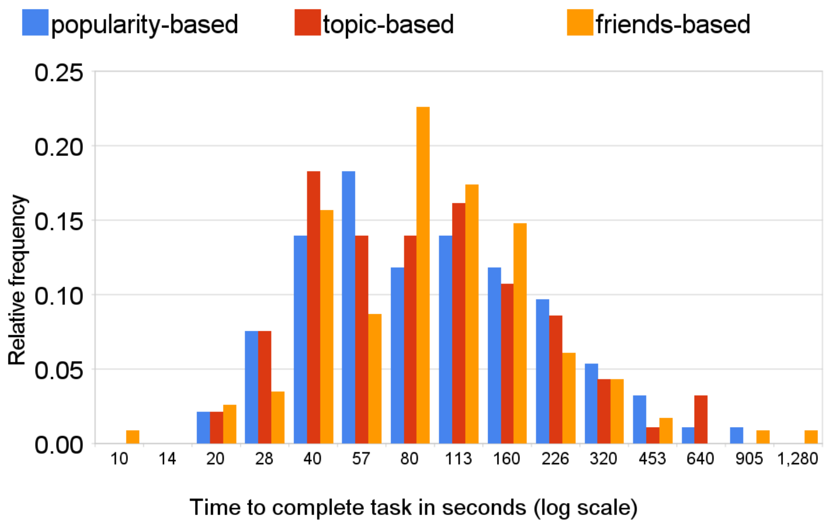

Figure 5.3. Histogram of time taking to complete tasks for different models.

of the participants finished the evaluation, since the search task was harder than in the pilot study. Less results were given per query which forced people to use more precise queries.

Model	Started	Completed	Rel. Frequency (%)
Popular	144	30	20.8 ±3.4
Topic	160	32	20.0 ±3.2
Social	148	37	25.0 ±3.6

Table 5.2. Number of completed tasks per model. While the social model appears to slightly outperform the other models, the difference is not statistically significant at the 5% confidence level.

Results in Table 5.2 show our social model slightly outperforming the popular and topic models. The results are not statistically significant.

To complete the tasks participants used multiple tags in their queries, a total of 54 for the popular model, 66 for the topic model and 68 for the social model. This suggests that the social model proposes tags that are more closely related to each other and therefore enables the user to make longer queries.

5.4 User survey

We conducted a short user survey together with the pilot study. Table 5.3 gives the statements that were asked to be ranked on a Likert scale. Figure 5.4 represents the answers of the participants for each question.

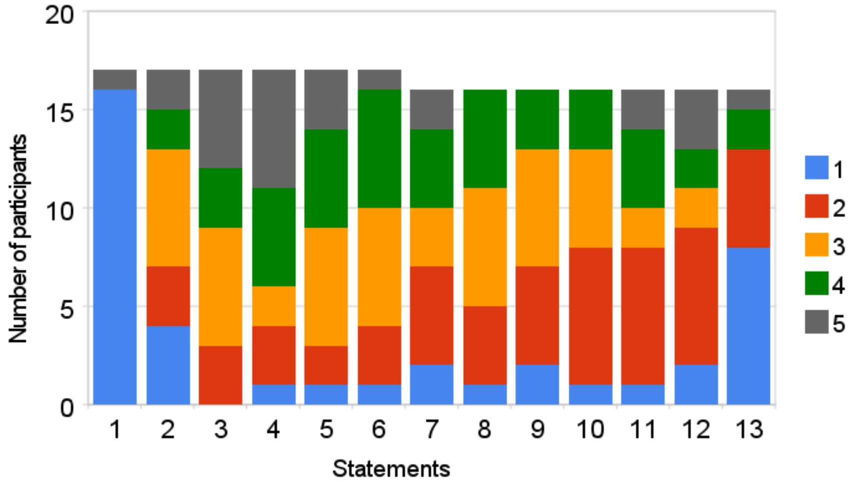

Figure 5.4. Number of participants per statement (best viewed in colour).

The answers to question 1[5] clearly shows that our users are heavy internet users which you would expect when conducting a survey in a computer science faculty. Eleven participants mostly disagree with statement 4 and 8 with statement 3 which are both statements about the usage of tagging systems, which shows that tagging is still a feature that is not broadly used by people even in a computer science department. Answers to statements 5 to 9 are inconclusive, participants are mostly undecided. No participant strongly disagree with statement 8 but only 5 mostly agree, finding items by navigating a tag cloud is a hard task for a human which shows that improvements regarding searchability are needed. Eight participants agree with statements 10 and 11 and 9 with statement 12. These three statements are about using the tag cloud to navigate various resources.

Most participants find it easy to navigate the tag cloud and would use a tag cloud to navigate the Web or their personal files. Eight participants out of 17 agree with the 13th statement, 13 mostly agree. This confirms the fact that tag-based navigation improves discovery of new resources.

[5]This is hard to follow so please have a look at Table 5.3 while reading this paragraph.

	Entry
1.	I use the internet regularly
2.	I regularly search for music online
3.	I often use tagging systems to search for information
4.	I often tag items in tagging systems
5.	I prefer to navigate tagging systems by clicking on tags rather than searching (via keyword queries)
6.	I am interested in popular music
	Exit
7.	I like navigating the tag cloud
8.	I think it is easy to find items by navigating the tag cloud
9.	I find that managing the selected keywords is easy
10.	I think I can find items quickly with the tag cloud
11.	I would use the tag cloud to navigate the web
12.	I would use the tag cloud to navigate files on my personal computer
13.	I think that tag cloud navigation helps with discovering new music

Table 5.3. Study statements

5.5 Experiment 2: Piloted recommendation

5.5.1 Purpose

The purpose of this experiment is to evaluate the effectiveness[6] of the models presented in the previous chapter when generating a tag cloud for piloting a recommendation for discovery of new and relevant items.

5.5.2 Method

The recommendation experiment consisted of tasks in which participants had to select a tag[7] from the tag cloud and then listen to a song recommended from the current query (the query being composed of the tags selected so far), participants would rate the song (whether they like it or not) and then go back to the new tag cloud generated according to the query and the model.

5.5.3 Results

Table 5.4 shows that the popular model outperforms the topic model and social model to generate tag clouds that lead participants to recommended songs that they like. This

[6] I measure the effectiveness in an application in which users give relevance feedback to recommendations based on which tags the users clicked.

[7] Any tag of interest to the participant.

Model	Rated	Liked	Rel. Frequency (%)
Popular	131	90	**68.7** ±4.1
Topic	104	60	57.7 ±4.8
Social	148	75	50.7 ±4.1

Table 5.4. Relative frequencies of liked ratings. The popular model significantly outperforms the other models at the 5% confidence level (according to the two-proportion unpooled one-sided z-test).

can be simply explained. Popular items are liked by the majority of people. It is most likely that if we recommend a popular song, it will be liked.

Model	Liked	Unkown&Liked	Rel. Frequency (%)
Popular	90	16	17.8 ±4.0
Topic	60	22	**36.7** ±6.2
Social	75	23	**30.7** ±5.3

Table 5.5. Relative frequencies of unkown resources within liked ratings. Both the topic and social models tend to lead the user to find more unknown music that they like than the popular model. Results are statistically significant at the 5% confidence level.

If we look at the relative frequencies of songs that were new to the participants within the songs that they liked, we find that the popular model is the least efficient, intuitively popular items are liked and already known, which is why they are popular because so many people know them. Table 5.5 shows that the topic model is the best model followed closely by the social model, both models outperform quite significantly the popular model. These results support our thesis that using social relationships enhances the recommendation of new and relevant information. The topic model performs better than the social model, we believe that once the social model is personalized, *i.e.* uses the actual social network of the participant instead of an overall probability from a social network, the social model would perform even better. This is what we look at in the following chapters.

5.6 Conclusion

Our work has some limitations, the number of participants of the pilot study and follow-up study is relatively small (17 and 20 participants) which does not allow us to draw strong conclusions. We focused our attention on only one dataset from Last.fm with online music data, the conclusions can not be generalised to tag cloud based navigation of other corpora.

Our survey shows that search is not practical with tag clouds whereas recommendation and discovery of new information is. Our follow-up study shows that in the case of recommendation of items that people liked and were new to them, the topic and social models perform much better than the popularity model. We can retain the following conclusions :

- Tag navigation is not practical for search.[8]
- People discover items when navigating the cloud.
- Experimenting with recommendation shows good results for the social model.

Based on the results of this work, the following chapter goes deeper in recommendation for discovery using social relationships.

[8] We have seen that the participants in the experiment had a hard time finding the tracks that were asked to search for, especially in the follow up study of the first experiment where the number of tracks shown was limited to ten, which forced the participants to specify the search using multiple tags.

Chapter 6

Design of a Social Application

In this chapter I describe the rationale behind the realization of a social media, Starnet. I follow the development cycle to explain what observations lead to choosing Facebook as a platform and music videos as a mean to experiment with the discovery of music in a social context. I define functionalities and relate them in an architecture with components of the system built. I give details of the technical implementation of the Facebook application. I discuss lessons learned and the difficulties and advantages of making a scientific experiment as a social application.

Contribution This Chapter is a contribution to **Web Engineering** and **Social media**. The methodology described to develop a whole system going from an observation of a phenomenon in an existing system to the evaluation of a newly developed one is new. The application of this methodology on the social phenomenon of music sharing in social networks is new.

6.1 Introduction

In this chapter I detail how Starnet was built following a development cycle that I had experienced and developed in the NEPOMUK European project. This cycle was developed to ease the communication between technical people on the system side and designers and user study people on the requirements side. In our case, the project is not so big therefore communication is not an issue, but following the steps of the development cycle gives a good explanation of the rationale behind the final system. Figure 6.1 gives a representation of the cycle, decomposed into two parts, the requirements and system each composed of steps. Going from one step to another can be made by forward or reverse engineering depending on what is the task at the moment and what has been modified. One can draw a parallel between the development cycle and a scientific cycle where functionalities are hypotheses and the implementation is an experiment in which to test them.

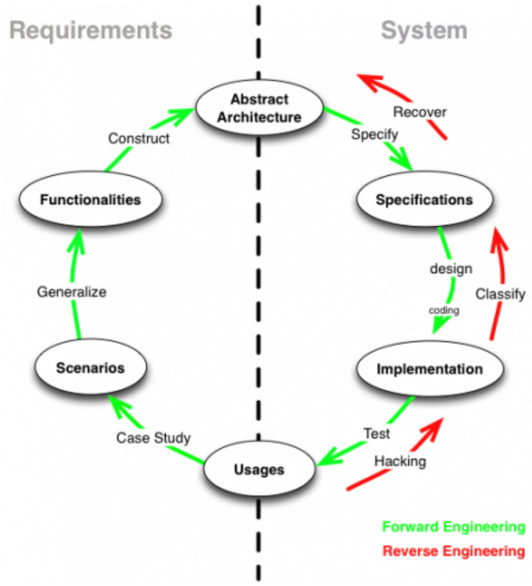

Figure 6.1. Software development cycle.

The cycle shows the steps of the making of a software system and how one derives from another. I speak of a development cycle as the development of a software system is never finished and each of its components are continuously evolving. Clockwise the cycle starts from the observation of particular **usages** that people make of an existing software system or expect from a new one. Abstracting from these observed usages, the software developer writes **scenarios** that are helpful to reason about the system, abstracted once more the scenarios help in deriving **functionalities**. The extracted functionalities are linked to components of the **specifications** in the abstract architecture. The components are then implemented to build the actual system.

The chapter outline follows the clockwise sequence of the development cycle as described above. It deepens on the implementation to give details of how the system is programmed.

6.2 Requirements : a system that fosters social interaction

A social web system is a system that extends social interaction. In this section, I describe what lead me to create Starnet, by looking at usages observed on the Facebook social network. These usages lead to various possible scenarios that I abstract in system functionalities that we want the system to cover.

6.2.1 Usages : observing the posting of links to your wall

Figure 6.2 shows a link to a music video posted by myself on Facebook. This is a common behavior on Facebook. People post music videos because they like the video, they want to share a feeling that this video express, share a new music track that they just discovered or recall a song they enjoyed with their friends.

Figure 6.2. A post of a music video on the Facebook network.

These posts made on Facebook are then published on the social channel. The friends of the person who made the post see the post in their social feed and just by clicking on the image of the video they play it. Of course people post other types of videos, not just music. For our experiment and software we reason only on music videos.

When other people see the post they can say they like it by clicking on the "like" button or type in a comment which might lead to a discussion.

The social interaction is complete when the posting of the music video leads to events outside of the system, in the material world. For instance people might speak together afterwards of the new album for which one posted the video. This effect could be measured by conducting a survey on people's behavior towards social recommendations.

6.2.2 Scenarios : Stories of social recommendation

The following scenarios are here to illustrate the observations made previously. For the scenarios I will use three fictional personas, namely Alice, Bob and Charlie.

Scenario 1 Alice and Bob have been going out for a month now. When Bob is at work he posts a song expressing how he misses Alice. Alice listen to the song and enjoys this attention from Bob so she expresses in the system that she likes the post. When meeting after work at the pub, Bob tells Alice how happy he was that she liked his post.

Scenario 2 Charlie just heard there is a new album from Radiohead. He finds the video of the single on Youtube and posts it to his wall. Alice discovers the news by seeing Charlie's post and directly listens to the track.

Scenario 3 Bob is nostalgic of when he had a band with his friends and posts a Led Zeppelin song on his wall that he was playing when he was a teenager. Charlie leaves a comment remembering the good times.

The three scenarios show different aspect of what the simple usage described previously might express or lead to. In the next subsection we extract a set of functionalities that cover the scenarios.

6.2.3 Functionalities : what do we expect from the system?

The functionalities should abstract from the scenarios what a user might expect from the software system to fill. This is useful in describing the system to someone newly introduced to and to relate the parts of the implementation to its usages. I list here the three major functionalities of the system and relate them to the scenarios.

Discover A user discovers music new to her that she likes. (Scenario 2)

Rate A user rates a recommendation of a music she likes or not. (Scenario 1)

Social interaction A user creates a social interaction with his friends. (Scenario 1,3)

In the following section we discuss how these functionalities are implemented in Starnet as a new system inspired from the observations made in this section.

6.3 System : Starnet, measuring music discoveries

In this section we follow a forward engineering thread from the cycle discussed in the introduction. We start by relating components to the expected functionalities into an abstract architecture, we then design the specifications of the system, and go to a lower level of abstraction to discuss implementation details of the system.

6.3.1 Abstract architecture : from functionalities to components

Functionalities	Components
Discover	Recommenders, Social feed
Rate	Rating Form, Video player
Social interaction	People votes, Social feed

Table 6.1. Architectural components.

Table 6.1 relates functionalities to components of the specifications. The discover functionality is covered by the recommenders and the social feed. This is what I evaluate in the next Chapter, how much each recommender helps the user in discovering new music she likes. The social feed brings in the database new music posted by people which leads to recommendations and therefore potential discoveries. Users rate by looking at the video of a track and using the rating form of the user interface which proposes a system of voting by 5 stars. Social interaction is provided by showing which people have voted or fed the system with the song, which gives the user a pointer to potential new friends or the information on the taste of his own friends.

In the following subsection we look at how these components interact and describe them in more details.

6.3.2 Specifications : what is the system supposed to do?

The specifications should give enough precisions such that someone can implement it. We first look at the design decisions and then give a description of each part of the system.

Design decisions

Starnet is implemented as a Facebook application. This means that a user is able to access the application from her Facebook account and that the system has access to the user data from Facebook. We observed in the requirements that Facebook users show this behavior of making music recommendations by posting links to music videos on their Facebook wall, therefore designing the system as a Facebook application enables us to enhance this behavior and gather links posted on the users' walls.

The system recommends music videos and enables the user to rate how she likes the video and if it is a discovery. This part is the major component of the scientific experiment. We will see that by choosing randomly a recommender and by enabling the user to rate the recommendation we can measure and analyse as shown in the previous chapter how a recommender affects the discovery and recommendation capacity of a system.

System components

Figure 6.3 shows the technical design of the system. We distinguish three parts in the system, namely the **user interface**, the **back end** and the **data sources**. The user interface is programmed in Php to ease the integration with the Facebook platform and renders an HTML page which contains Javascript to deal with events, for instance of the video player. The back end is programmed as a set of Ruby scripts which interact with the data sources, either in SQL with the database or by calling application programming interfaces through REST calls and interpreting the JSON returned.

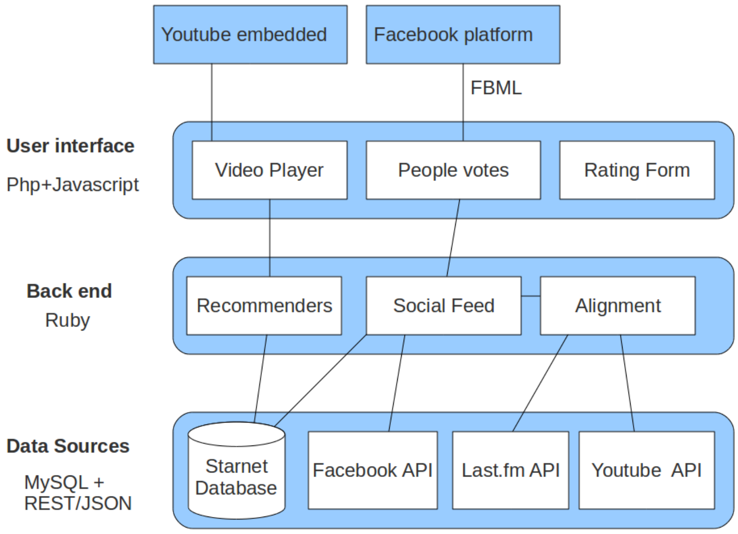

Figure 6.3. Interaction of the components of Starnet.

We now look at each component and describe their role in the system.

Video player The player plays the recommended video by embedding a Youtube video into the Starnet page. This video player is piloted by a Javascript library and enables us to catch events. For instance a relevant event to be caught is when the video is no longer available, *i.e.* we had in the database a video identifier on Youtube but Youtube does not allow the video to be played anymore. This can be due to multiple reasons, for instance the video can be removed by the Youtube user who posted it, or the artist or music label has asked Youtube to remove it for copyright reasons. When this happens we catch the event and recommend seamlessly another video. We also pilot the player to autostart when the page is loaded and play a new song once the song is finished playing. Pressing the next or bail buttons of the rating form also pauses the video and stops downloading it.

People Votes When a video is recommended and it has been reviewed by other people we display the people and their votes in the interface. This is done in FBML, the Facebook Markup Language, which is Javascript enabling to fetch user data such as their names and pictures at the level of the user interface. In the database we only store the facebook user id.

Rating Form The rating form is composed of 5 stars whose interaction behavior is programmed in Javascript. If the user clicks on a star then this is the value selected for the rating and the stars below and including this one are highlighted, if she clicks again this puts the value to 0 and displays all the stars in gray. There are two buttons in the form, "Next" which stores the rating and recommends a new track and "Bail" which sets the rating to 0 and recommends a new track. The "Bail" button came later on in the development of the application to simplify the action when a music video is really horrible, the user just wants it to stop playing so the button stops the video and if the rating was set to something other than 0 by default it sets it to 0, preventing the user to have to click first on the stars to put a 0 rating and then on "next".

Recommenders This is the core component of the system. As discussed in the previous chapter, the social shuffle principle is to select randomly a recommender and then apply this recommender to make a recommendation. Here the recommender can either be complete random on the dataset, a social recommendation or a non-social recommendation. The social recommendation selects randomly a track from the top tracks rated by friends of the user, non including the ones she has already rated. The non-social recommender applies the same principle on top tracks of people who are not the user's friends. Selecting the recommender randomly makes the ratings suitable for further scientific analysis. We store which recommender was used when the user rating is stored. The recommenders are Ruby methods which interact with the Starnet database.

Social Feed The social feed runs as a cron job on the server, it is not called directly by the user interface as its execution is too long. It interfaces with the Facebook API to maintain the database by fetching changes of the social network, if any user has new friends, and looking at newly posted videos. The new videos are then processed by the alignment component to determine if it is a music video and what song it is. The remaining videos are added to the database and automatic ratings are made for the user who posted the link. The rating stored is a 5 stars rating considering the user knows the song, these ratings are not analysed in the previous chapter and stored with a value for the recommender set to "social feed".

Alignment Once a Youtube link is fetched from the Facebook social network, we isolate the Youtube id of the video and use the Youtube API to get informations about it. We keep only the videos identified as being music by the Youtube API. The Youtube title of a video requires some alignment to figure out which is the artist and track name in the database. We split the title using different heuristics depending on how it is written and extract the artist and track name, then we use the Last.fm API to match the artist name and track name and check if it is an actual song and if it is correctly written.

Starnet Database The database stores user information and their relationships from their friends on Facebook. Currently we store more than four thousand people, most of them being friends of people registered in the application. The tracks identified with a Youtube video from previous work account for 240 thousand. Most of the calculation of the recommenders is done in the database and runs as queries. These queries have to run fast as the rendering of the page depends on how long it takes to make a recommendation.

Facebook API We interact with Facebook both at the top level of the application to authenticate the user and at the bottom level to fetch data. For this purpose we use the **Graph API** which gives access to the social graph. To use the graph API, the application server needs to authenticate for the application and the user and gets an access token if the user has given access to the application. Accessing the data of the social graph is made by making an HTTP call to an address of the form `https://graph.facebook.com/cedric.mesnage` which returns JSON data by default. We then parse the data and update the database if necessary. We frequently update the database to maintain the social network by looking at the list of friends of each registered user and the list of posted links of all users in the database.

Last.fm API The Last.fm API is used to get information about the tracks. Here again we use REST/JSON communication to interact with their API, all data is available once you have a registered developer key. In previous work we explored their social network and fetched songs information for the top songs of 300 thousand people which led us to a database of a million songs. This database acts a seed. For new songs, once the song is recognized we get the genre of the song by fetching the top tags of the artist. The tags are free form labels given by users, the tags for a song might not be very abstract, therefore looking at the top tags of an artist gives better results.

Youtube API The Youtube API enable us to check whether a song in our database has a music video. The other way around is when we have a new link posted in the network, we check if that video is a music video, if so we get its title and add it to the database. We also fetch the tags given to the video, but looking at it closely, they give worst results than tags from Last.fm.

In the next subsection we look in more details at the implementation of the application as a Facebook application.

6.3.3 Implementation details

Designing a Facebook application is similar to designing a stand alone Web application. The implementation architecture is a standard model-view-controller[68]. The user management is provided by Facebook which requires on our side to deal with the authentication of the user through Facebook. In this section we give details on how to create a Facebook application and how the user accesses it through her account. We explain how the authentication of the user works with OAUTH 2.0 authentication and discuss how to integrate with the Facebook Graph API.

Creating and configuring a Facebook application

The application runs on our server and is displayed in a **canvas** in the Facebook platform. The canvas is actually an IFrame in HTML. An IFrame is a visual part of a Web page which contains another Web page. When the Facebook page of the application is rendered on the client, the IFrame interacts with our server to authenticate the user and the application.

Figure 6.4 shows the Facebook platform and in gray the location of the IFrame canvas for the application. The page called to fill the content of the canvas is programmed in Php and generates an HTML and Javascript page. The Facebook platform gives the user facebookid in the HTTP session and challenges the application with its application secret. If the application is not able to complete the challenge by communicating with the Facebook API, the application is not displayed.

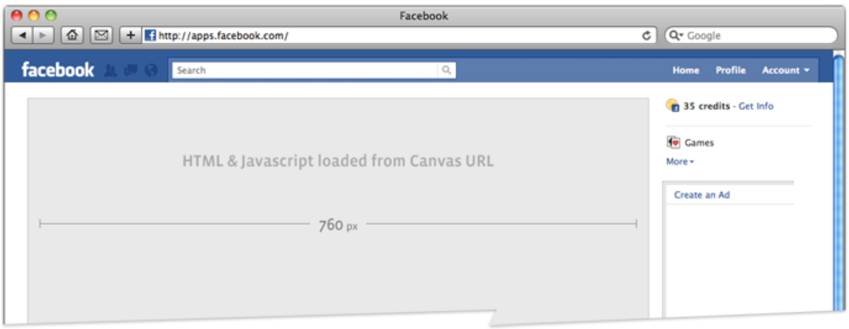

Figure 6.4. Facebook application canvas.

The space allocated to the application is of 760 pixels wide and any height. This is a constraint that has to be taken into account when designing the user interface as for instance the videos we display as embedded from Youtube are the smallest embed that can be made and already take 400 pixels wide.

When creating a Facebook application you must be registered as a Facebook developer. Within the Developer application you can create and manage your Facebook applications. After creating an application and setting up its details and description, we configure the integration page of the application settings as shown in Figure 6.5.

The core settings are set by Facebook and will be used in the code to authenticate the application when interfacing with the API or when loading the canvas page. The canvas page url is the URL of the application where users can reach the application, here `http://apps.facebook.com/starnet_app`. The canvas URL is the URL of the actual

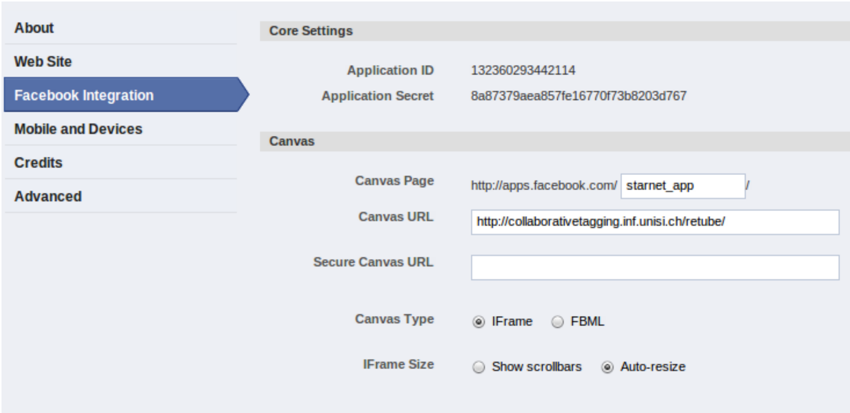

Figure 6.5. Facebook integration page.

server that will process the application, in our case `http://collaborativetagging.inf.unisi.ch/retube`, this server is hosted at the university of Lugano, the name retube comes from a former version of the application which purpose was to republish Youtube videos.

Once the application is created it is accessible through the Facebook platform. We then develop the application on the server at the address of the canvas URL. This makes it a difficult development process in my opinion as it requires to develop completely the application online, especially for testing. I will discuss this problematic aspect from a software development point of view later on.

Accessing the application

Facebook users can find the application by searching for it in the Facebook search or by browsing the application directory. Starnet is listed under entertainement. Another way of finding the application is by looking at the applications installed by the friends of the user. When the user looks at the applications, she sees the most used applications of her friends. When registering to the application the user is prompted with a registration agreement allowing the application to access the user's data.

Figure 6.6 shows my sidebar of Facebook. Under the application list Starnet is displayed with its icon. By clicking on the Starnet application, the Facebook page of the application `http://apps.facebook.com/starnet_app` is loaded. This makes the application well integrated within the Facebook framework. The user is shown the link anytime she logs in Facebook, giving us more chances that she will go often to check the application.

The more an application is used, the more it gets displayed to other users as a potential application they might want to use. This viral means of promoting an application is interesting from an information diffusion point of view. Once the application has grown enough we could analyse how people came to know the application.

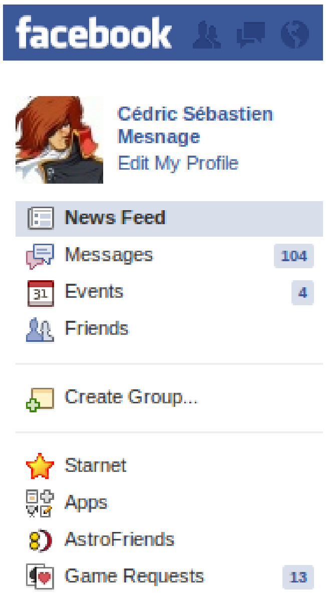

Figure 6.6. Facebook sidebar of a Starnet user.

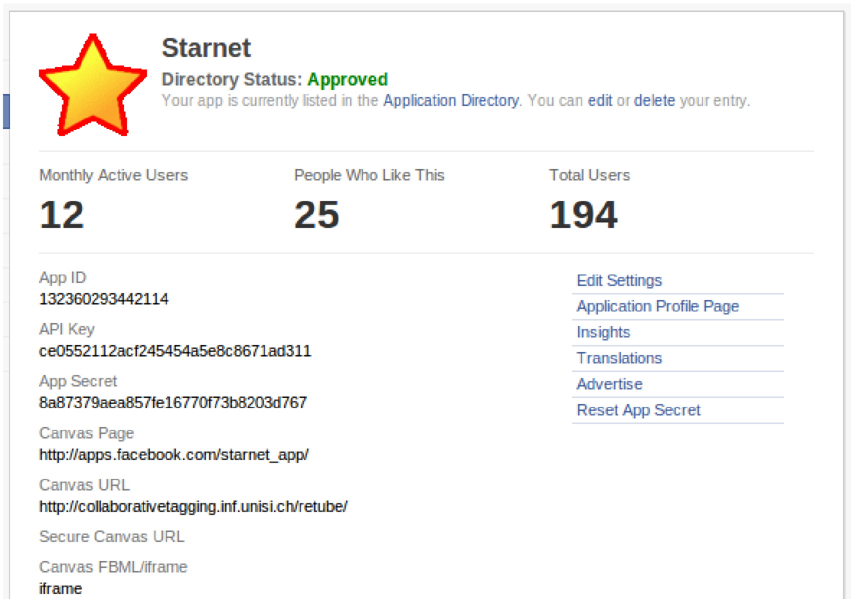

Figure 6.7. Description of Starnet from the developer application.

Figure 6.7 is a screenshot of the overview of Starnet from the developer application. The insights also give data about the usage of the application, such as the number of users over time and the number of new users over time. Here we can see that Starnet has 194 users and that 25 people liked the application. The monthly users, here 12, is the number of users who used the application over multiple months. In the database only 50 people made at least one rating, which means that a fourth of the users who registered actually made a rating the others did not continue to use the application.

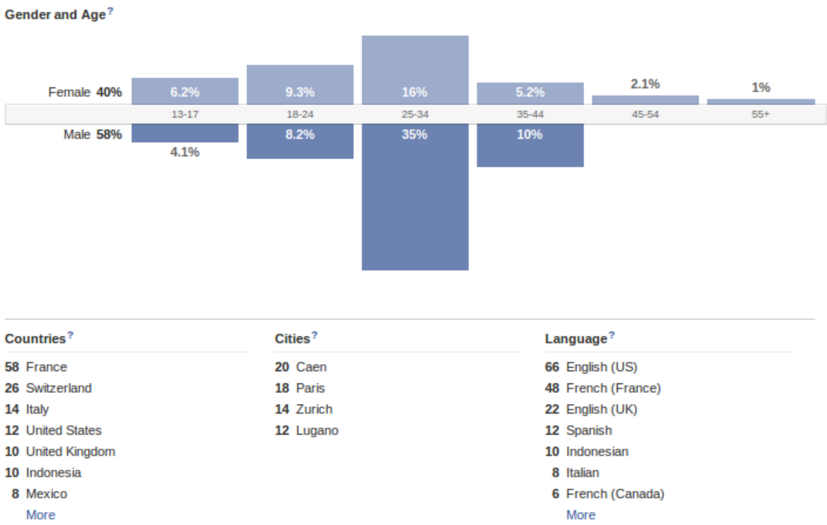

Figure 6.8. Demographics of the total installed users of Starnet.

Figure 6.8 is a screenshot from the demographics analysis of Starnet users given by Facebook. Most users are in the 25-34 age range. The gender repartition is 40% female and 58% male. The countries from which the users originate are mainly France, Switzerland, USA, UK and indonesia making Starnet a truly international application.

Working with the Facebook API

When authenticating the application for the current user, interacting with the Facebook API works fine as it is one API call. I found that for fetching friends data using the Graph API is quite slow. It can take up to 20 secondes for one user, which certainly prevents us from doing it each time a user access the application. Even running it as a cron job for all users is problematic as currently we hold more than 4000 Facebook users (the application users and their friends) and it takes about 5 hours to update the network and the links they posted.

Using FQL queries, the Facebook query language, with the old REST API speeds up the process a bit, but Facebook informs us that they are not going to maintain this API and recommends to use the Graph API.

Even the display of Facebook users in the application is slow (getting their name and picture). All in all working with the Facebook API is problematic.

6.4 Conclusion

In this chapter I gave the rationale behind creating a social application experiment. We followed the software development cycle. The observation of posting behavior on Facebook led to the idea of building a social feed. The potential functionalities of a system based on social recommendations are discover, rate and social interaction. I presented the architecture and design of the experiment and its relations with external services. I discussed the implementation of the application and the interaction with external APIs and the creation of a Facebook application. In the following chapter I evaluate the *discover* functionality of the system by testing the hypothesis that social recommendations lead to more discoveries.

Chapter 7

Evaluating Discovery

I show in this chapter that music discoveries are diffused through the social network of its listeners. I describe the methodology used to build an experiment in the form of a music video recommendation application in Facebook. I give results of the experiment as a statistical analysis which shows the validity of the hypothesis.

Contribution This Chapter is a contribution to **Social media**. Music recommendations which are usually the result of a collaborative filtering process in web-based systems are improved in terms of discovery if the recommendations come from the user's social network. The evaluation of such a system is new.

7.1 Introduction

This chapter provides an evaluation for the *discover* functionality of the Starnet system presented in the previous chapter. To evaluate this functionality we test the main hypothesis on which the system is built, the fact that recommendations made from someone's social network lead to better discoveries. As shown in Table 7.3 the results are statistically significant at the 0.005 level, 45% of social recommendations are discoveries.

7.2 Hypothesis

My hypothesis is that :

> Music recommendations in a web-based application such as Starnet lead to more discoveries if they come from the user's social network.

7.3 Design of the experiment

To test the hypothesis described previously we set up the recommendation strategy as described later in this chapter such that recommendations are taken randomly from a pool of music tracks and based on user feedback we either recommend the discoveries to users friends or non-friends. We therefore have two conditions, either the recommendations come from the user's social network or they do not. We measure one independant variable, the user's rating for a music track, and perform repeated measures, each user evaluates social and non-social recommendations multiple times. The experiment being non-parametric, we can perform a Wilcoxon test for the statistical hypothesis test.

The sample of users are young people (students from the University of Lugano Switzerland and the University of Caen, France) who use internet applications.

7.4 Methodology

A discovery is when a subject likes a track that she has never heard before. Discoveries are random and naturally diffused through the social network. The experiment reproduces this process so that we can analyse the response of people to recommendations.

The recommendations are either randomly picked from the pool of tracks of the data set or collaboratively from the tracks recommended by other people. The data set is built from Last.fm crawling described in the previous chapters. For each of the million tracks of this dataset, I searched on the youtube API for a corresponding video, tagged with the term "Music". Using this process I collected 250 thousand music videos for unique tracks. Each time a track is recommended, the subject of the experiment rates the track on a 0 to 5 likert scale visually represented as stars and tells if she knows the track already or not. A collaborative recommendation is either social or non-social, *i.e.* a friend of the subject gave it a high rating or a non-friend gave a high rating.

I give a statiscal analysis of the ratings which shows that social recommendations are better recommendations than non-social recommendations.

To test the hypothesis, I pose the problem as a recommendation problem. The recommendation problem is defined as follows: given a pool of items, choose an item that the subject has not seen and that is relevant to her. A collaborative recommendation makes use of previously rated items by other subjects to choose the item to be recommended. If collaborative recommendations issued to a subject on items rated by people from the social network of the subject lead to more successful recommendations than on items rated by people not in the subjects social network, then we show that social recommendations are more appropriate as collaborative recommendations.

7.4.1 Experimental setting

In the previous study on tag navigation I learned that people were frustrated that they could not listen to the whole song when they liked it (Last.fm restricts the playback to only a piece of the song). That is why I chose to use music videos from Youtube. As a matter of fact, people currently use Youtube music videos to share music on the internet, by sending links by email or within social networks. In this section I describe how I built the dataset which serves as pool of tracks for the recommender, the application built as a recommender system and the selection of subjects.

Dataset The dataset built from Last.fm contained a million tracks which were fetched through their API. I had explored a piece of the social network, about 300 thousand people and their relationships. For each user I fetched their tag profile, the list of tags they used to organize songs, giving us a set of more than 300 thousand unique tags. For each tag the API allows to fetch the top 50 most popular tracks tagged with this tag. Only the set of tracks is relevant for this study. On the Youtube API, I searched for each track, videos which had in their title the artist name and the track title. I restricted to only the videos which were tagged with "Music" and selected the most popular video for each track. Using this process about a quarter of the songs from Last.fm had a video on Youtube, about 252 thousand tracks.

Facebook application Starnet is a Facebook application[1] fed by ratings on random selections. A positive rating spawns diffusion through the user's social network, ie the shuffle recommendation becomes social. The interface (Figure 7.1) consists of the current track description (its title and artist name), the music video associated, a tag cloud of the user's profile, a rating form consisting of 5 stars and a "Next" and Bail button(sets the stars to 0 and votes).

Subjects I choose to make a within subject study. Each subject acts as her own control group, by getting random recommendations. The subjects are originally from my own social network. It extended as people were invited by the ones who joined the application. The experiment gathered 68 people out of which 31 used the system and did more than 10 evaluations. These 31 people made 4966 ratings in about 4 months (from the 29th June 2010 to the 18th October 2010).

7.4.2 Recommendation strategy

The goal of the recommender system is not to make the best recommendation possible but to reproduce a setting where we can test the hypothesis. A better recommendation system would make use of the genres (or tags) of the tracks previously rated by the

[1] `http://apps.facebook.com/starnet_app`

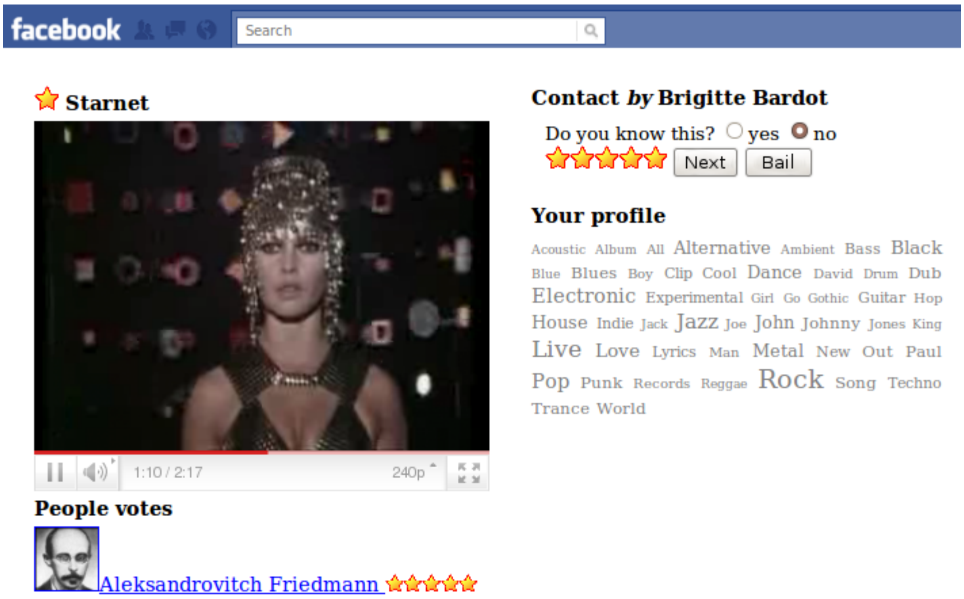

Figure 7.1. Screenshot of Starnet on Facebook.

subject. In our situation either the recommendation is random, social, or non-social. The Table 7.1 describes how the recommender is selected where $r1$ and $r2$ are random numbers between 0 and 1, α and β are numbers between 0 and 1 both set to 0.5, these could be tune later on.

$r1 < \alpha$	$r1 \geq \alpha$	
Random	Collaborative	
	$r2 < \beta$	$r2 \geq \beta$
	Non-social	Social

Table 7.1. Social diffusion recommendation methods.

Random The random selection is a query which selects the tracks that have not been rated by the subject and orders them by random numbers.

Social the social recommender selects a track randomly from the tracks that have been rated by friends of the subject with a rating superior to 2 stars.

Non-social the non-social recommender selects a track randomly from the tracks that have been rated by people who are not friends of the subject with a rating superior to 2 stars.

Collaborative selects a track randomly from the set of tracks that have been rated with a rating above 2 stars and not yet rated by the subject.(Social or non-social)

7.4.3 Statistical method

To conduct a statistical analysis of the subjects ratings, I generated histograms of the relative frequencies in percentage for various interests (number of stars per ratings). We look here at the various measurements and methods.

Wilcoxon test To test the hypothesis significance, I perform a one-tailed Wilcoxon test. The experiment is non-parametric as the tested variable is the user's ratings which are from 0 to 5, so we can rank the data but the data is not rational. The two conditions of the repeated measurements that I compare are the social and non-social provenance of the recommendations. I do not compare with random as we will see later on, the random recommendations are much less efficient than the social or non-social ones.

Discoveries A measurement of the success of a recommendation model to discover new and relevant tracks for a subject is the ratio of already discovered tracks and all rated tracks. Figure 7.2 represents how these sets relate for a particular user.

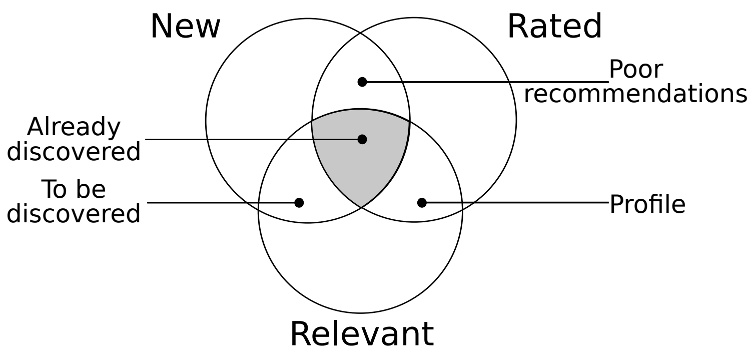

Figure 7.2. Discovery recall diagram.

Histogram Histograms enable to quickly compare quantities for multiple values of a variable. In this case we look at the number of stars on the x axis and the corresponding percentage of ratings on the y axis. Each bar corresponds to a particular set of ratings, for instance the ratings coming from random or collaborative recommendations or ratings on songs known or unknown to the subject.

Heat map Heat maps enable to look at relation between the values of two variables on a grid. I use heat maps to look at how the ratings of a subject relates to ratings on the same songs from friends and non friends.

Precision and recall The precision and recall in the context of recommendation are different than in the context of information retrieval. The precision is simply the number of good recommendations over the number of ratings given, the recall is the number of good recommendations over the estimated size of what could have been recommended. The estimate is calculated by random sampling.

7.5 Results

In this section I analyse the results from the analysis of the ratings made by the subjects of the experiment. I compare in various histograms the social and non-social recommendations leading to ratings of tracks known or unknown. I will show that social recommendations give better ratings than non-social ones.

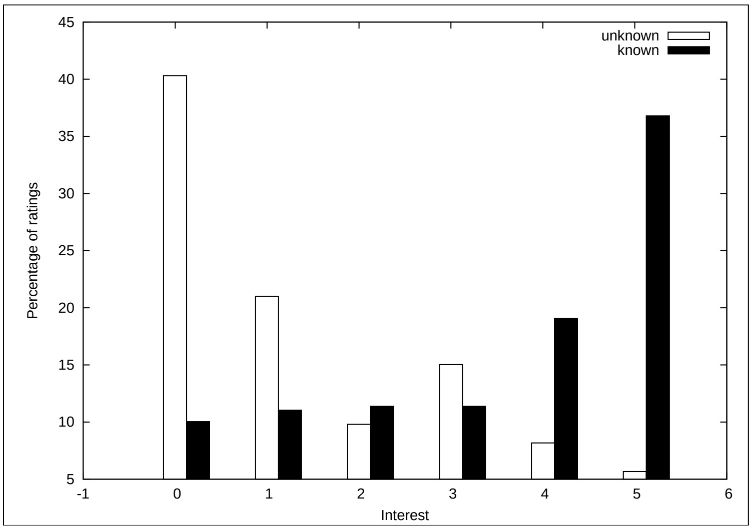

Figure 7.3. Histogram of percentage of ratings for known and unknown per interest.

Figure 7.3 gives us a first look at the ratings. It includes all the ratings per interest value. I divided the ratings between the ones where the subject specified she knows the track and the ones specified as unknown, respectively represented as black and white bars. A bar represent the proportion of ratings of the corresponding interest within the total of ratings of that kind (known or unknown). 36.78% of the tracks known by people get 5 stars and 19% 4 stars. Below 4 stars, each interest gets equally 11% of the known tracks. 67.21% of the known tracks get 3 stars or more. Most of the unknown tracks are completely disliked by people (40.32% with 0 stars) and drops to 21% for 1 star and 9.8% for 2 stars. 3 stars seems to be a threshold for what is unknown as it goes up to 15.02%, 28.87% of unknown tracks are rated with above 3 stars. This globally shows an inversion between known and unknown tracks, most unknown tracks are disliked whereas most of known tracks are liked. What interests us is to increase the number of recommended unknown tracks that are liked. In this figure we looked indifferently at all the recommenders, we should differentiate between collaborative and random recommendations.

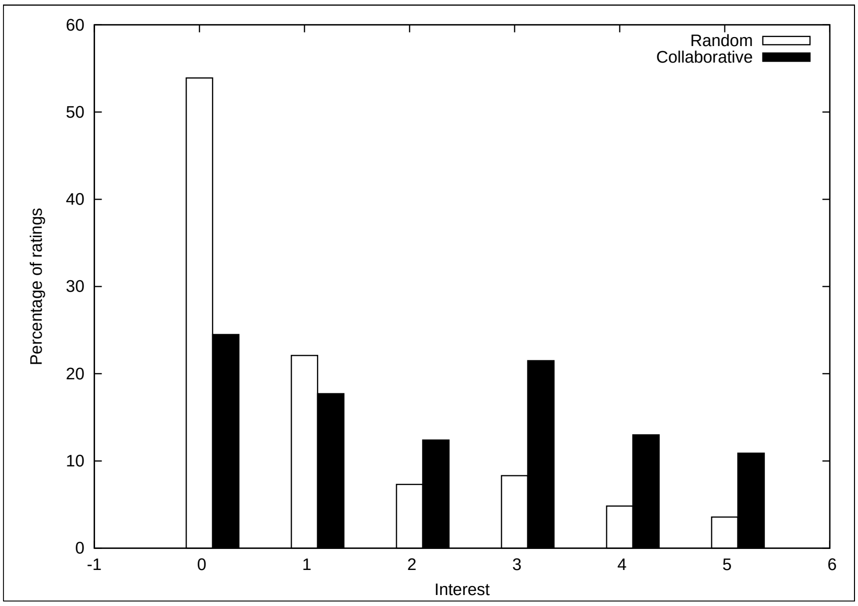

Figure 7.4. Histogram of percentage of ratings for collaborative and random ratings per interest.

Figure 7.4 represents the proportion of collaborative and random ratings per interest value. 53.9% of random recommendations get 0 stars and 22.09% 1 star, so 75.99% of random recommendations get less than 2 stars. The remaining 24% are spreaded from 7.3% for 2 stars to 3.56% for 5 stars. In comparison 10.89% of collaborative recommendations lead to 5 stars ratings and 21.49% for 3 stars; 45.38% collaborative recommendations get 3 stars or more against 16.7% for random recommendations. The percentage of 0 stars drops to 24.49% for collaborative recommendations and 42.2% get less than 2 stars against the 75.99% for random recommendations. In this visualization, the threshold of 3 stars is more evident in collaborative recommendations. This is due to the way collaborative recommendations are made as from the random recommendations only the ones which get a rating superior to 2 stars are then used as collaborative recommendations. We now look only at the collaborative ratings dividing them between social and non-social recommendations.

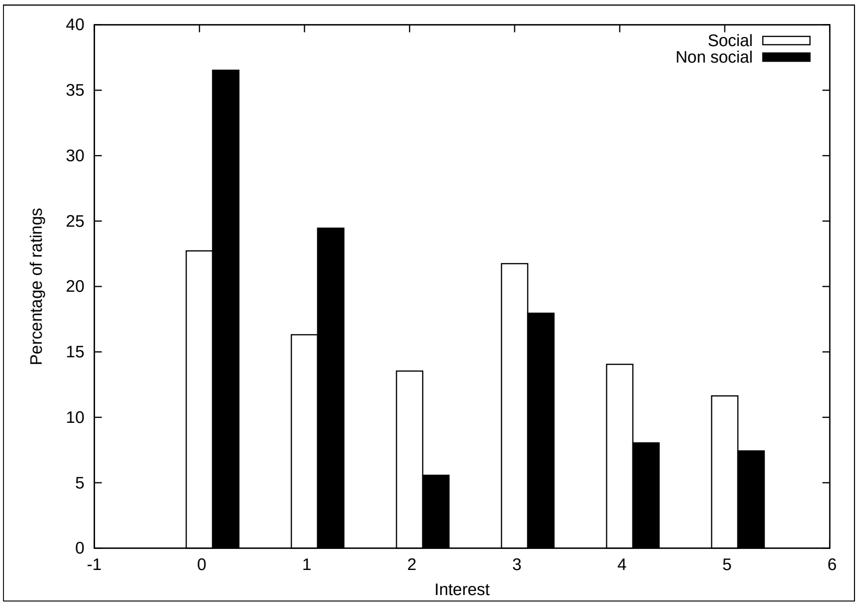

Figure 7.5. Histogram of percentage of social and non-social ratings per interest.

Figure 7.5 represents the ratings interests of the collaborative recommendations the white bars are social recommendations, recommendations coming from random recommendations and spread to friends of the one who gave it a high rating. The black bars are ratings coming from non-social recommendations, recommendations that were made to participants who were not friends with subjects who already rated this track. The social and non-social ratings are differently spread other the interests so subjects react differently to recommendations coming from their friends or from people they do not know. 66.6% of non-social recommendations get less than 3 stars with a clear outlier on 2 stars ratings with 5.57%, this is probably explained by the fact that the taste of people who are not friends are more extreme and therefore either liked or disliked. 33.43% of non-social recommendations get 3 stars or more against 47.43% for social recommendations, the difference is even clearer above 4 stars, 15.47% for non-social recommendations and 25.69% for social ones. In this visualization we see that social recommendations tend to lead to better ratings, but we mix known and unknown tracks.

Most of the ratings state that the song is unknown to the subject, that is why the left histogram on Figure 7.6 is quite similar to the histogram of Figure 7.5. The same conclusions apply, the difference on 5 stars ratings is less evident as 9.29% of social recommendations lead to 5 stars unknown ratings against 6.5% for non social recommendations. In general, the social recommendations lead to more unknown good recommendations(45.86% above 3 stars against 33.87%) than the non social ones and less bad recommendations(40.24% less than 2 stars against 59.88%). The histogram on the right shows ratings where the subject specified she knew the track. 36.1% of social rec-

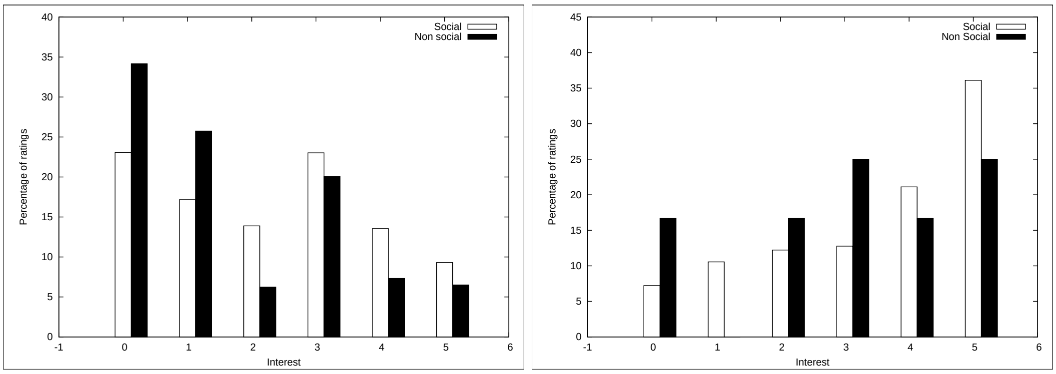

Figure 7.6. Histograms of percentage of social and non social ratings per interest on tracks unknown on the left and known on the right.

ommendations lead to 5 stars ratings and 21.1% to 4 stars against 25% and 16.6% for the non social ones, we can say that songs people know are liked.

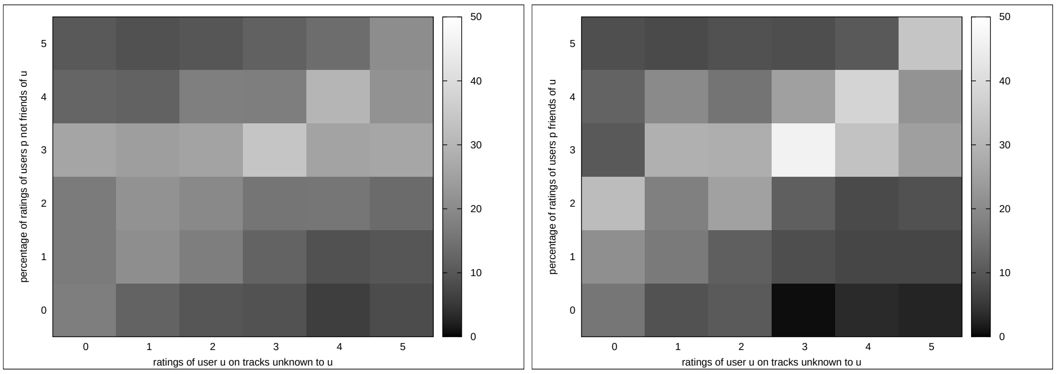

Figure 7.7. Heat map of ratings made on the same tracks, by people who are not friends and did not know the song on the left and people who are friends on the right.

Figure 7.7 shows two heat maps. Each square represents the proportion of ratings made by other people on the same tracks for each interest value. For instance the lower left square is the percentage of tracks rated 0 by people who are not friends with the ones who rated the same tracks with 0 other the total number of ratings made on these tracks, here 15.78%. The left hand heat map represents the relation between ratings of people who are not friends and the right one the relation between ratings of people who are friends. We notice that the diagonal is whiter on both heat maps, meaning that globally people agree on ratings on the same tracks. The social heat map is more contrasted, showing that friends agree more with their ratings on same tracks than people who are not friends. 46% of ratings made by friends on tracks rated 3 rate them 3, 37.82% for

(4,4) ratings and 33.94% for (5,5) against 33.7% for (3,3) made by people who are not friends, 29.7% for (4,4) and 20.63% for (5,5).

		Random		Non social		Social	
Rating	Estimate	Precision	Recall	Precision	Recall	Precision	Recall
> 0	105629	0.4489	0.0106	0.75	0.0003	0.6761	0.0042
> 1	53097	0.2256	0.0106	0.5192	0.0005	0.4842	0.0060
> 2	36339	0.1544	0.0106	0.5769	0.0008	0.4137	0.0075
> 3	16851	0.0716	0.0106	0.2115	0.0006	**0.2503**	0.0099
> 4	6590	0.0280	0.0106	0.0576	0.0004	**0.1454**	0.0147

Table 7.2. Precision and recall results for discovered tracks with random, non social and social recommenders.

Table 7.2 shows the precision and recall results for different interest values. The estimate number of tracks to discover used to compute the recall is based on the random sampling, basically the random precision times the number of tracks in the database. The precision above low values (from 0 to 2) is higher for non social recommendations than social ones. What is relevant here is the precision values for high ratings (above 3 and 4) which shows that social recommendations give a higher precision of good recommendations than the non social one. The recall values are quite low because they include the estimate of the total number of tracks to discover in the dataset and the experimentation did not explore all the dataset.

7.6 Statistical significance

To prove the validity of my hypothesis I perform a standard statistical hypothesis test. As discussed in previous sections, the data produced by the experiment is non-parametric and I performed repeated measures for each user over two conditions, either the recommendation is social, or non-social. Table 7.3 shows the signed rank Wilcoxon T statistics for our data. The top 10 users of the system who performed more than 200 single evaluations.

Wilcoxon T statistics summary for the top 10 users who produced more than 200 evaluations each. The T value is of 3 and for a one-tailed test and $n = 10$ in the table of critical values of the Wilcoxon T statistics the critical value for p=0.005 is 3. The null hypothesis can therefore be rejected a the 0.005 level.

For the Wilcoxon test, the mean positive ranked difference score and the mean negative ranked difference score could be reported to convey the effect size, 0.0839 is the negative difference mean and 1.0162 is the positive difference mean.

The result of this test rejects the null hypothesis, which means that the hypothesis that *music discoveries are social* is true for our sample within the system and can be

Social	Non-social	Difference	Rank	Signed Rank
2.9176	3.0	-0.0824	1	-1
2.4259	2.5114	-0.0855	2	-2
3.24	2.8721	0.3679	3	3
2.7701	2.1463	0.6238	4	4
3.434	2.7143	0.7197	5	5
1.4974	0.7244	0.7730	6	6
3.41	2.4259	0.9841	7	7
3.028	1.6056	1.4224	8	8
3.1642	1.6818	1.4824	9	9
2.7765	1.0196	1.7569	10	10

Table 7.3. Wilcoxon T statistics summary for the top 10 users who produced more than 200 evaluations each. The T value is of 3 and for a one-tailed test and $n = 10$ in the table of critical values of the Wilcoxon T statistics the critical value for p=0.005 is 3. The null hypothesis can therefore be rejected a the 0.005 level.

generalized to the population of young internet users.

7.7 Discussion of the experiment

In this section I discuss aspects of the experiment which are questionable in terms of scientific concerns.

Score by default I tried different values for the default score, when a video is first displayed to a user. At first I had put just one star, this lead to many tracks rated with one whereas what people really meant was zero or no vote at all. I tried to use the average of the votes by other people which is meaningful from a least effort point of view are people are more likely to want the average as a vote. The problem again is that we can not tell if users really meant to vote for the average or just left it without looking. The solution here is to make the vote explicit by an AJAX call, not stopping the playback of the video as opposed to store the vote when "next" is clicked.

The bail button At first I did not have a bail button. It was requested by users who wanted to quickly stop the playing of a video. When using the random recommender most tracks are not liked by people. Therefore one button to make the 0 vote and switch to the next track is meaningful.

Rating with 5 stars In previous experiments I had another way of rating discoveries, *i.e.* "I dislike", "I am indifferent" and "I like". In this experiment I chose to use a five star rating as it is what is seen in many websites. This also allows to make a more continuous analysis of the ratings. After seeing the users use the system with 5 stars, I realize that a 0 to 5 value is subjective. People associate different meanings to each value. The solution here would be to have an explicit likert scale with terms that remove the subjectivity of the number associated to it.

Displaying previous votes In the user interface the votes made by people previously on the video played are displayed to the current user. This poses the question if the vote of people is affected by this or not. When a user sees that one of his friend has given a high rating to a video, even if she dislikes it she might decide to give it a different rating from what she actually feels towards the video. On the other hand as we analyse the social aspect of the recommender, displaying the vote is part of the experiment. The fact that there is a social interaction created by the recommendation has to be measured and if we chose not to display the voters, this social interaction would not exist. The solution here is to randomly choose some recommendations that would not display the previous voters so that one can compare and measure the effect of displaying voters, them being friends of the user or not.

7.8 Conclusion

In this chapter I presented an experiment in social diffusion. I described the methodology and gave statistical results. The analysis of the results lead us to the following conclusions.

- Recommended tracks that were known are mostly liked whereas recommended tracks that are unknown are mostly disliked.
- Collaborative recommendations lead to more good recommendations than random recommendations.
- Social recommendations give more good recommendations and less bad recommendations than the non social recommendations.
- People have more taste in common with their friends than with people who are not their friends, especially on what they like.

Result on discovery As shown in Figure 7.6 45% of social recommendations are discoveries, this is an amazing result. Half of what people recommend to each other is new and relevant, this is higher than what we found in previous chapters where we had 30% of discoveries using the social model that was not personalized.

These conclusions tend to confirm that social diffusion is a good mechanism for recommendation and discovery. The statistical results of precision and recall show that the problem is a hard problem which needs further research.

Chapter 8

Conclusion

In this thesis I presented my work in tag navigation, search and social recommendation. My main conclusion is the fact that we can use social relationships as a means to enhance information recommendation and discovery, mapping the natural phenomenon of social recommendation with software. The use of random selection and social recommendation results in a serendipitous process which I call the "social shuffle".

8.1 Summary of the Research

The processes of human music discovery on the Web can be decomposed into two forms :

- the user browses a music collection and make discoveries of new and relevant music by exploration.
- the system makes recommendations to the user.

In this thesis I argue that for both of these processes the use of social network data enhances the user experience in discovering new music she likes.

For browsing I experimented with social tagging and navigation through a tag cloud. I give formal definitions of bayesian models to generate tag clouds using social relationships and applied them in live Web-based experiments to compare with models based on popularity and topic models. I conducted three user evaluations to compare the models for search and recommendation of music with social network data gathered from Last.fm. My survey shows that search with tag clouds is not practical whereas recommendation is promising. I reported statistical results and compared the performance of the models in generating tag clouds that lead users to discover songs that they liked and were new to them. I find statistically significant evidence at 5% confidence level that the topic and social models outperform the popular model.

For recommendation I experimented with a social application. I built a Facebook application that recommends music videos from Youtube based on the user social network. I gave a statistical analysis of the participants ratings which shows that social diffusion leads to more good recommendations. I compared social diffusion with non social recommendation and random recommendation and gave precision and recall for the three models for music discovery for each rating value, I found that for 5 stars ratings of songs that were not known to the participant the social model clearly outperforms the non-social and random models. I gave a detailed rationale of the implementation of the social application.

The exploration of these two aspects of music discovery and the fact that for both aspects the use of social relationships is crucial in making good recommendations gives us confidence in supporting the argument of the thesis.

8.2 Limitations

This work has some limitations which I discuss here.

8.2.1 Subjectivity of ratings

The way people rate music is subjective. In the social application experiment I observed that people have different rating behaviors, some are extreme and rate mostly with 0 and 5 star ratings, some others are more widespread and give ratings across all possible values. The perception of a song by a person is clearly subjective, there is no objective way of judging of the value of a song. On the other hand the only way to assess the value of a recommendation is to ask people to rate that recommendation.

8.2.2 Subjectivity of social network data

The social network of a person is the result of a subjective process. People meet and connect through rather random and unpredictable events and the fact that they share a social relationship is a matter of perception and personal taste. Thus using social network data to make recommendations is controversial. It would be interesting to compare the results in making recommendations from general social network data such has Facebook and specialized social network data for music in which people connect because they share common music interests. My intuition is that the latter would result in better recommendations and less discoveries.

8.2.3 Empirical evaluation

This work is empirical in the sense that the hypothesis is tested against experimental data. The problem with empirical evaluations is that it does not lead to a theoretical proof. We can only support the thesis, the experimental data shows that the thesis is verified under my experimental setting. In fact, the results given depend as well on the systems built to test my hypotheses, I tried, as much as possible to reduce the system effect by limiting the functionalities provided.

8.3 Directions

These directions are as much for you than for me. You are welcome to take them and go away.

- Implement a social recommender which takes distance in the social network into account (friend of friend of friend).
- Implement a recommender which trains itself using relevance feedback from the listener to explore a social network.
- Align Youtube videos with MusicBrainz.
- Interview World Music producers, artists, festival and world music night organizers.
- Identify socio-cultural bridges.
- Produce inter-cultural music.
- Serendipity, the faculty of making happy discoveries by accident.

8.4 Conclusion

It is hard to conclude on such a long work and when writing becomes so difficult. Probably the easiest is to start by recollecting the main topic of this thesis, the hypotheses, list its contributions, look at its strengths and weaknesses and recall the main conclusions.

The topic of **Music Discovery** came about during the work, at first my interest was on tagging, it drifted slowly towards music, first by using music data for experiments, at that time I was listening heavily to my mp3 player in shuffle mode during the first experiment on tag navigation trying to do search I realised, thanks to the participants and the experiment's exit survey that they were discovering new music while trying to complete the experiment, the experiment itself was a failure in many ways, but I had found myself a topic.

My two main hypotheses are :

- Tag Navigation performs better if the tag relations are drawn from a social network.

- Music recommendations lead to more discoveries if they come from as user's social network.

The contributions are mainly orientated around two concepts, tag navigation and social diffusion. With both concepts the goal was to enhance serendipity, therefore the amount of music discovered by accident. First I define the two concepts. For tag navigation I give bayesian models to generate tag clouds and an end-user evaluation with three web-based experiments. For social diffusion I give a simple algorithm and an evaluation through a web-based experiment in a social network. The definition of the **discover** functionality and its evaluation. These contributions are based in the fields of social media, web engineering and tag navigation.

The strength of this thesis is to be using end-user experiments to test the hypotheses, to confront itself with the real world. Its strength is also its weakness as the experiments are small, at the end it is only a PhD work. It would be interesting to integrate this work within large commercial plaforms to experiment with many more users.

People discover items when navigating a tag cloud. Experimenting with recommendation gives statistically significant good results for social based navigation.Social recommendations lead to more discoveries than non-social ones. People have more taste in common with their friends than with people who are not their friends, especially on what they like.

Bibliography

[1] P. Aman and L. Liikkanen. Survey of music recommendation aids. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.

[2] L. Backstrom, R. Kumar, C. Marlow, J. Novak, and A. Tomkins. Preferential behavior in online groups. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 117–128, New York, NY, USA, 2008. ACM.

[3] Z. BAKER. *Social Networking and Music: the impact of myspace as a Digital Platform for Discovery and Distribution*. Bachelor's thesis, Cardiff School of Journalism, Media and Cultural Studies, Cardiff University, 2009.

[4] L. Barrington, R. Oda, and G. Lanckriet. Smarter than genius? human evaluation of music recommender systems. In *International Symposium on Music Information Retrieval*. Citeseer, 2009.

[5] A. Bernardi, S. Decker, L. Van Elst, G. Aastr, T. Groza, C.S. Mesnage, M.Jazayeri, et al. *The Social Semantic Desktop A New Paradigm Towards Deploying the Semantic Web on the Desktop*. IGI Global, October 2008.

[6] D. Bogdanov, M. Haro, F. Fuhrmann, E. GÃşmez, and P. Herrera. Content-based music recommendation based on user preference examples. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.

[7] P. Raghavan C. D. Manning and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[8] M. J. Carman, M. Baillie, and F. Crestani. Tag data and personalized information retrieval. In *SSM '08: Proceeding of the 2008 ACM workshop on Search in social media*, pages 27–34, New York, NY, USA, 2008. ACM.

[9] C. Cattuto. Semiotic dynamics in online social communities. *The European Physical Journal C - Particles and Fields*, 46(0):33–37, aug 2006.

[10] C. Cattuto, V. Loreto, and L. Pietronero. From the Cover: Semiotic dynamics and collaborative tagging. *Proceedings of the National Academy of Sciences*, 104(5):1461, 2007.

[11] C. Cattuto, V. Loreto, and L. Pietronero. From the Cover: Semiotic dynamics and collaborative tagging. *Proceedings of the National Academy of Sciences*, 104(5):1461, 2007.

[12] O. Celma. *Music Recommendation and Discovery in the Long Tail*. PhD thesis, Universitat Pompeu Fabra, Barcelona, 2008.

[13] E.H. Chi and T. Mytkowicz. Understanding the efficiency of social tagging systems using information theory. In *HT '08: Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pages 81–88, New York, NY, USA, 2008. ACM.

[14] S.J. Cunningham, D. Bainbridge, and D. McKay. Finding new music: a diary study of everyday encounter with novel songs. In *Proceedings of the 8th International Conference on Music Information Retrieval, Vienna, Austria*, pages 83–88, 2007.

[15] B. Fields, K. Jacobson, C. Rhodes, M. d'Inverno, M. Sandler, and M. Casey. Analysis and exploitation of musician social networks for recommendation and discovery. *Multimedia, IEEE Transactions on*, PP(99):1–1, 2009.

[16] B. Fields, C. Rhodes, and M. d'Inverno. Using song social tags and topic models to describe and compare playlists. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.

[17] J. Fokker, J. Pouwelse, and W. Buntine. Tag-Based Navigation for Peer-to-Peer Wikipedia. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh*, 2006.

[18] J. Freyne, R. Farzan, P. Brusilovsky, B. Smyth, and M. Coyle. Collecting community wisdom: integrating social search & social navigation. In *IUI '07: Proceedings of the 12th international conference on Intelligent user interfaces*, pages 52–61, New York, NY, USA, 2007. ACM Press.

[19] M. Gaffney and P. Rafferty. Making the long tail visible: social networking sites and independent music discovery. *Program: electronic library and information systems*, 43(4):375–391, 2009.

[20] R. Garg, M.D. Smith, and R. Telang. Discovery of music through peers in an online community. *Hawaii International Conference on System Sciences*, 0:1–10, 2011.

[21] S. Golder and B.A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[22] S. Golder and B.A. Huberman. Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, 32(2):198–208, 2006.

[23] T.L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101, 2004.

[24] T. Groza, S. Handschuh, K. Moeller, G. Grimnes, L. Sauermann, E. Minack, C.S. Mesnage, M. Jazayeri, G. Reif, and R. Gudjonsdottir. The nepomuk project - on the way to the social semantic desktop. In Tassilo Pellegrini and Sebastian Schaffert, editors, *Proceedings of I-Semantics' 07*, pages pp. 201–211. JUCS, 2007.

[25] H. Halpin, V. Robu, and H. Shepherd. The Dynamics and Semantics of Collaborative Tagging. In *SAAW'06, Semantic Authoring and Annotation Workshop*, 2006.

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

[27] M.J. Halvey and M.T. Keane. An assessment of tag presentation techniques. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1313–1314, New York, NY, USA, 2007. ACM.

[28] T. Hammond, T. Hannay, B. Lund, and J. Scott. Social Bookmarking Tools (I). *D-Lib Magazine*, 11(4):1082–9873, 2005.

[29] Y. Hassan-Montero and V. Herrero-Solana. Improving tag-clouds as visual information retrieval interfaces. In *InScit2006: International Conference on Multidisciplinary Information Sciences and Technologies*, 2006.

[30] M. A. Hearst and D. Rosner. Tag clouds: Data analysis tool or social signaller? In *HICSS '08: Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, page 160, Washington, DC, USA, 2008. IEEE Computer Society.

[31] P. Herrera, Z. Resa, and M. Sordo. Rocking around the clock eight days a week: an exploration of temporal patterns of music listening. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.

[32] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, New York, NY, USA, 2008. ACM.

[33] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. BibSonomy: A social bookmark and publication sharing system. In *Proceedings of the Conceptual Structures*

Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures, pages 87–102, 2006.

[34] T. Ishikawa, P. Klaisubun, and M. Honma. Navigation efficiency of social bookmarking service. pages 280–283, Nov. 2007.

[35] M. Jazayeri, Mesnage C.S., and J. Rose. *Modern Web Application Development*, pages 131–147. John Wiley & Sons, Inc., 2007.

[36] O. Kaser and D. Lemire. Tag-cloud drawing: Algorithms for cloud visualization. In *Tagging and Metadata for Social Information Organization Workshop, WWW07*, 2007.

[37] R.M. Keller, S.R. Wolfe, J.R. Chen, J.L. Rabinowitz, and N. Mathe. A bookmarking service for organizing and sharing urls. In *Selected papers from the sixth international conference on World Wide Web*, pages 1103–1114, Essex, UK, 1997. Elsevier Science Publishers Ltd.

[38] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 195–202, New York, NY, USA, 2009. ACM.

[39] R. Lambiotte and M. Ausloos. Collaborative tagging as a tripartite network. *LECTURE NOTES IN COMPUTER SCIENCE*, 3993:1114, 2006.

[40] P. Lamere. Social tagging and music information retrieval. *Journal of New Music Research*, 37(2):101–114, 2008.

[41] P. Lamere and D. Eck. Using 3d visualizations to explore and discover music. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, 2007.

[42] A. Laplante. *Everyday life music information-seeking behaviour of young adults: an exploratory study*. PhD thesis, McGill University, Montreal, Canada, 2008.

[43] A. Laplante. The role people play in adolescents' music information acquisition. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.

[44] K. Lee, W. S. Yeo, and K. Lee. Music recommendation in the personal long tail: Using a social-based analysis of a user's long-tailed listening behavior. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.

[45] T. W. Leong, F. Vetere, and S. Howard. The serendipity shuffle. In *OZCHI '05*, pages 1–4. CHISIG, 2005.

[46] M. Levy and K. Bosteels. Music recommendation and the long tail. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.

[47] M. Levy and M. Sandler. Music information retrieval using social tags and audio. *Multimedia, IEEE Transactions on*, 11(3):383–395, 2009.

[48] R. Li, S. Bao, Y. Yu, B. Fei, and Z. Su. Towards effective browsing of large scale social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 943–952, New York, NY, USA, 2007. ACM.

[49] A. Lillie. *MusicBox: Navigating the space of your music*. PhD thesis, Massachusetts Institute of Technology, 2007.

[50] M. Lux, M. Granitzer, and R. Kern. Aspects of broad folksonomies. *Database and Expert Systems Applications, 2007. DEXA '07. 18th International Conference on*, pages 283–287, Sept. 2007.

[51] M. Magas, M. Casey, and C. Rhodes. mhashup: fast visual music discovery via locality sensitive hashing. In *ACM SIGGRAPH 2008 new tech demos*, pages 1–1. ACM, 2008.

[52] C. Marlow, M. Naaman, D. Boyd, and M. Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, New York, NY, USA, 2006. ACM.

[53] C. Marlow, M. Naaman, D. Boyd, and M. Davis. Position Paper, Tagging, Taxonomy, Flickr, Article, ToRead. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, May 2006.

[54] C.S. Mesnage, R.P. Brixtel, and M.J. Carman. Serendipituous social shuffle. In *WOMRAD 2010 1st Workshop on Music Recommendation and Discovery*, 2010.

[55] C.S. Mesnage and M. Carman. Piloted search and recommendation with social tag cloud-based navigation. In *WOMRAD 2010 1st Workshop on Music Recommendation and Discovery*, 2010.

[56] C.S. Mesnage and M.J. Carman. Tag navigation. In *Proceedings of the 2nd international workshop on Social software engineering and applications*, pages 29–32. ACM, 2009.

[57] C.S. Mesnage and M.J. Carman. Piloted search and recommendation with social tag cloud-based navigation. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.

[58] C.S. Mesnage and M. Jazayeri. Specifying the collaborative tagging system. In *SAAW'06, Semantic Authoring and Annotation Workshop*, 2006.

[59] C.S. Mesnage and M. Jazayeri. Social thinking to design social software: A course experience report. In *1st International Workshop on Social Software Engineering and Applications*, 2008.

[60] C.S. Mesnage and Michele Lanza. White coats:web-visualization of evolving software in 3d. In *Vissoft'05*, 2005.

[61] C.S. Mesnage and E. Oren. Extending ruby on rails for semantic web applications. *Web Engineering*, pages 506–510, 2007.

[62] C.S Mesnage, A. Rafiq, S. Dixon, and R. Brixtel. Music discovery with social networks. In *WOMRAD 2011 2nd Workshop on Music Recommendation and Discovery*, 2011.

[63] D.R. Millen and J. Feinberg. Using social tagging to improve social navigation. In *Workshop on the Social Navigation and Community based Adaptation Technologies*, 2006.

[64] D.R. Millen, J. Feinberg, and B. Kerr. Dogear: Social bookmarking in the enterprise. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 111–120, New York, NY, USA, 2006. ACM Press.

[65] S. Miller, P. Reimer, S. Ness, and G. Tzanetakis. Geoshuffle: Location-aware, content-based music browsing using self-organizing tag clouds. In *International Symposium on Music Information Retrieval*, 2010.

[66] E. Oren, A. Haller, M. Hauswirth, B. Heitmann, S. Decker, and C.S. Mesnage. A flexible integration framework for semantic web 2.0 applications. 24:64–71, 2007.

[67] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina. Clustering the tagged web. In *Second ACM International Conference on Web Search and Data Mining (WSDM 2009)*, November 2008.

[68] T. Reenskaug. Models - views - controllers. Technical report, Technical Note, Xerox Parc, 1979.

[69] G. Reif, T. Groza, S. Handschuh, C.S. Mesnage, M. Jazayeri, and R. Gudjonsdottir. Collaboration on the social semantic desktop. In *Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS 2007) at CAiSE*. Citeseer, 2007.

[70] A. W. Rivadeneira, Daniel M. Gruen, Michael J. Muller, and David R. Millen. Getting our head in the clouds: toward evaluation studies of tagclouds. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 995–998, New York, NY, USA, 2007. ACM.

[71] V. Sandvold, T. Aussenac, O. Celma, and P. Herrera. Good vibrations: Music discovery through personal musical concepts. In *Proceedings of 7th International Conference on Music Information Retrieval (to be published), Victoria, Canada*. Citeseer, 2006.

[72] P. Schmitz. Inducing ontology from flickr tags. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, May 2006.

[73] S. Sen, F.M. Harper, A. LaPitz, and J. Riedl. The quest for quality tags. In *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work*, pages 361–370, New York, NY, USA, 2007. ACM.

[74] S. Sen, S. K. Lam, A.M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F.M. Harper, and J. Riedl. tagging, communities, vocabulary, evolution. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 181–190, New York, NY, USA, 2006. ACM Press.

[75] U. Shardanand and P. Maes. Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217. ACM Press/Addison-Wesley Publishing Co., 1995.

[76] J. Sinclair and M. Cardew-Hall. The folksonomy tag cloud: when is it useful? *J. Inf. Sci.*, 34(1):15–29, 2008.

[77] F. Smadja, A. Tomkins, and S. Golder. Collaborative web tagging workshop. In *WWW2006, Edinburgh, Scotland*, 2006.

[78] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W. Lee, and C. L. Giles. Real-time automatic tag recommendation. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 515–522, New York, NY, USA, 2008. ACM.

[79] M. Sordo, F. Gouyon, and L. Sarmento. A method for obtaining semantic facets of music tags. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.

[80] S. Stober and A. Nürnberger. A multi-focus zoomable interface for multi-facet exploration of music collections. In *Proc. of 7th Int. Symposium on Computer Music Modeling and Retrieval*, 2010.

[81] S.J. Tepper and E. Hargittai. Pathways to music exploration in a digital age. *Poetics*, 37(3):227–249, 2009.

[82] B. Tomasik, J.H. Kim, M. Ladlow, M. Augat, D. Tingle, R. Wicentowski, and D. Turnbull. Using regression to combine data sources for semantic music discovery. In *International Symposium on Music Information Retrieval*. Citeseer, 2009.

[83] J. Trant. Social classification and folksonomy in art museums: Early data from the steve. museum tagger prototype. *Proceedings of the 17th SIG Classification Research Workshop*, 2006.

[84] J. Trant and B. Wyman. Investigating social tagging and folksonomy in art museums with steve.museum. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, May 2006.

[85] D.R. Turnbull. *Design and development of a semantic music discovery engine*. PhD thesis, University of California, San Diego, 2008.

[86] D.R. Turnbull, L. Barrington, G. Lanckriet, and M. Yazdani. Combining audio content and social context for semantic music discovery. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 387–394. ACM, 2009.

[87] L. von Ahn. *Human Computation*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2005.

[88] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, May 2006.