Womrad 2010

Workshop on Music Recommendation and Discovery

September 26th, 2010

Barcelona, Spain

Copyright Information

[1] http://creativecommons.org/licenses/by/3.0/

Organizing Committee

Workshop Organizers

- Amélie Anglade - Centre for Digital Music, Queen Mary, University of London, UK
- Claudio Baccigalupo - Earbits, Los Angeles, CA, US
- Norman Casagrande - last.fm, London, UK
- Òscar Celma - BMAT, Barcelona, Spain
- Paul Lamere - The Echo Nest, Somerville, MA, US

Program Committee

- Mathieu Barthet (Centre for Digital Music, Queen Mary, University of London)
- Klaas Bosteels (last.fm)
- Sally Jo Cunningham (Waikato University)
- Justin Donaldson (Indiana University)
- Benjamin Fields (Goldsmiths, University of London)
- Emilia Gomez (Music Technology Group, Universitat Pompeu Fabra)
- Fabien Gouyon (INESC Porto)
- Peter Knees (Johannes Kepler University Linz)
- Neal Lathia (University College London)
- Daniel Lemire (Open Scholar, Montréal)
- Yves Raimond (BBC Audio & Music interactive)
- Markus Schedl (Johannes Kepler University Linz)
- Mohamed Sordo (Music Technology Group, Universitat Pompeu Fabra)

Preface

Welcome to WOMRAD, the Workshop on Music Recommendation and Discovery being held in conjunction with ACM RecSys.

In the last twenty years, there has been an amazing transformation in the world of music. Portable listening devices have advanced from the Sony Walkman that allowed you to carry ten songs in your pocket to the latest iPhone that can put millions of songs in your pocket via music subscription services such as Spotify or Rhapsody. Twenty years ago a typical personal music collection numbered around a thousand songs. Today, a music listener has access to millions of songs, drawn from all styles and genres from all over the world. The seemingly infinite choice today's music listener faces can lead to a rich music listening experience, but only if the listener can find music that they want to listen to.

Traditionally, music recommender systems have focused on the somewhat narrow task of attempting to predict a set of new artists or tracks for purchase or listening. Commerce sites like iTunes use music recommendation as a way to increase sales. Internet radio sites like Pandora use music recommendation as a way to offer personalized radio to millions of listeners. The success of music recommendation at iTunes and Pandora has led some to suggest that 'music recommendation is solved'. Indeed, for narrow use cases like improving sales in a mainstream music store, or for creating satisfactory personalized radio streams, music recommendation may be good enough. However, this does not mean that music recommendation is solved. As music listeners spend more time interacting with multi-million song music collections, the need for tools that help listeners manage their listening will become increasingly important. Tools for exploring and discovering music especially in the long tail, tools for organizing listening, tools for creating interesting playlists, tools for managing group listening will all be essential to the music listening experience. Music recommendation technologies will be critical to building these tools.

The WOMRAD workshop focuses on next generation of music recommender systems. Accepted papers fall into five categories:

- **Time Dependency** - 1 paper - explorations in temporal patterns of music listening
- **Social Tagging** - 3 papers - how semantic tags can be used to explain, compare and steer music recommender systems
- **Human-Computer Interaction** - 2 papers - how music listeners interact with music and music recommender systems
- **Content-based Recommendation** - 2 papers - techniques for recommendation base on audio content
- **Long Tail** - 2 papers - how can systems make effective recommendations of new or unpopular content

We are pleased to offer this selection of papers and hope that it serves as evidence that there is much interesting and fruitful research to be done in the area of music recommendation and discovery. We offer our thanks to all of the authors who submitted papers to this workshop.

The Organizers, October 2010

Keynote Presentation

The Dark Art: Is Music Recommendation Science a Science?

***Michael S. Papish**, Product Development Director, Rovi Corporation*

Music preferences are emotional, subjective and full of social and cultural meaning. Practical experience building industrial recommendation applications suggests that user "trust" (a fuzzy concept combining user psychology with UI design and presentation) often overshadows actual results. What if making good music recommendations is actually a Dark Art and not a foundational problem of Information Retrieval Science? By tracing the beginnings of MIR, we present an early attempt at a Philosophy of Recommendation Science which tries to answer:

- Does recommendation science exist only as a practical application?
- Is it possible ground-truth metrics such as those proposed in the ISMIR 2001 Resolution don't actually exist?
- What types of solvable scientific problems should receive academic attention from the MIR community?
- Cee Lo's Teeth: Scariest in the entire history of recorded music?

Table of Contents

Time Dependency

Social Tagging

Computer-Human Interaction

Content-based Recommendation

Long Tail

Rocking around the clock eight days a week: an exploration of temporal patterns of music listening

Perfecto Herrera
perfecto.herrera@upf.edu

Zuriñe Resa
Music Technology Group
Department of Technology
Universitat Pompeu Fabra
zuri_resa@hotmail.com

Mohamed Sordo
mohamed.sordo@upf.edu

ABSTRACT
Music listening patterns can be influenced by contextual factors such as the activity a listener is involved in, the place one is located or physiological constants. As a consequence, musical listening choices might show some recurrent temporal patterns. Here we address the hypothesis that for some listeners, the selection of artists and genres could show a preference for certain moments of the day or for certain days of the week. With the help of circular statistics we analyze playcounts from Last.fm and detect the existence of that kind of patterns. Once temporal preference is modeled for each listener, we test the robustness of that using the listener's playcount from a posterior temporal period. We show that for certain users, artists and genres, temporal patterns of listening can be used to predict music listening selections with above-chance accuracy. This finding could be exploited in music recommendation and playlist generation in order to provide user-specific music suggestions at the "right" moment.

Categories and Subject Descriptors
H.5.5 Sound and Music Computing – methodologies and techniques, modeling.

General Terms
Measurement, Experimentation, Human Factors.

Keywords
Music context analysis, Playlist generation, User modeling, Music metadata, Temporal patterns, Music preference.

1. INTRODUCTION
Among the requirements of good music recommenders we can point to, not only delivering the right music but, delivering it at the right moment. This amounts to consider the context of listening as a relevant variable in any user model for music recommendation. As existing technologies also make it possible to track the listening activity every time and everywhere it is happening, it seems pertinent to ask ourselves how this tracking can be converted into usable knowledge for our recommendation systems. Music listening decisions might seem expressions of free will but they are in fact influenced by interlinked social, environmental, cognitive and biological factors [21][22].

Chronobiology is the discipline that deals with time and rhythm in living organisms. The influence of circadian rhythms (those showing a repetition pattern every 24 hours approximately, usually linked to the day-night alternation), but also of ultradian rhythms (those recurring in a temporal lag larger than one day like the alternation of work and leisure or the seasons), has been demonstrated on different levels of organization of many living creatures, and preserving some biological cycles is critical to keep an optimum health [18]. The observation that human behavior is modulated by rhythms of hormonal releases, exposure to light, weather conditions, moods, and also by the activity we are engaged into [12][3] paves the way to our main hypothesis: there are music listening decisions that reflect the influence of those rhythms and therefore show temporal patterns of occurrence. The connection would be possible because of the existing links between music and mood on one side, and between music and activity on the other side. In both cases, music has functional values either as mood regulator [23] or as an activity regulator [13]. Therefore, as mood and activity are subject to rhythmic patterns and cycles, music selection expressed in playlists could somehow reflect that kind of patterning [26][23]. More specifically, in this paper we inquire on the possibility of detecting that, for a specific user, certain artists or musical genres are preferentially listened to at certain periods of the day or on specific days of the week. The practical side of any finding on this track would be the exploitation of this knowledge for a better contextualized music recommendation. Our research is aligned with a generic trend on detecting hidden patterns of human behavior at the individual level thanks, mainly, to the spread of portable communication and geolocation technologies [4][20].

2. RELATED RESEARCH
While recommendations based on content analysis or on collaborative filtering may achieve a certain degree of personalization, they do miss the fact that the users interact with the systems in a particular context [19]. Furthermore, several studies have shown that a change in contextual variables induces changes in user's behaviors and, in fact, when applying contextual modelling of the users (i.e., considering the time of the day, the performed activity, or the lighting conditions), the performance of recommendation systems improves both in terms of predictive accuracy and true positive ratings [8][25]. Although context-based music recommenders were available since 2003 [1], time information is a recently-added contextual feature [7][17].

WOMRAD 2010 Workshop on Music Recommendation and Discovery, colocated with ACM RecSys 2010 (Barcelona, SPAIN).

A generic approach to the characterization of temporal trends in everyday behavior has been presented in [10], where the concept of "eigenbehavior" is introduced. Eigenbehaviors are characteristic behaviors (such as leaving early home, going to work, breaking for lunch and returning home in the evening) computed from the principal components of any individual's behavioral data. It is an open research issue if Eigenbehaviors could provide a suitable framework for analyzing music listening patterns. A model tracking the time-changing behavior of users and also of recommendable items throughout the life span of the data was developed for the Netfix movie collection [14]. This allowed the author to detect concept drifts and the temporal evolution of preferences, and to improve the recommendation over a long time span.

Although research on behavioral rhythms has a long and solid tradition, we are not aware of many studies about their influence on music listening activities. The exception is a recent paper [2] where users' *micro-profiles* were built according to predefined non-overlapping temporal partitions of the day (e.g., "morning time slot",). The goal of the authors was to build a time-aware music recommender and their evaluation of the computed micro-profiles showed their potential to increase the quality of recommendations based on collaborative filtering. Most of that reported work was, though, on finding optimal temporal partitions. As we will see, there are other feasible, maybe complementary, options that keep the temporal dimension as a continuous and circular one by taking advantage of circular statistics. Developed forty years ago and largely used in biological and physical sciences, circular statistics has also been exploited in personality research for studying temporal patterns of mood [15][16]. To our knowledge, it is the first time they are used in the analysis of music-related behavior, though applications to music have been previously reported [5][9].

3. METHODOLOGY

3.1 Data Collection

Getting access to yearly logs of the musical choices made by a large amount of listeners is not an easy task. Many music playing programs store individual users' records of that, but they are not publicly accessible. As a workable solution, we have taken advantage of Last.fm API, which makes possible to get the playcounts and related metadata of their users. As raw data we have started with the full listening history of 992 unique users, expressed as 19,150,868 text lines and spanning variable length listening histories from 2005 to 2009. The data contained a user identifier, a timestamp, Musicbrainz identifiers for the artist and track, and a text name for the listened track.

The artist genre information was gathered from Last.fm using the Last.fm API method *track.getTopTags()*, which returns a list of tags and their corresponding weight[1].This list of tags, however, may relate to different aspects of music (e.g. genre, mood, instrumentation, decades...). Since in our case we need a single genre per track, we first clean tags in order to remove special characters or any other undesirable characters, such as spaces, hyphens, underscores, etc. Then irrelevant tags (i.e., those having a low weight) are removed and the remaining ones are matched against a predefined list of 272 unique musical genres/styles gathered from Wikipedia and Wordnet. From the genre tags we obtained for each song, we select the one with the highest weight. If there are several tags with the highest weight, we select the one with the least popularity (popularity is computed as the number of occurrences of a specific genre in our data-set).

3.2 Data cleaning

Data coming from Lastfm.com contain playcounts that cannot be attributable to specific listening decisions on the side of users. If they select radio-stations based on other users, on tags or on similar artists there are chances that songs, artists and genres will not recur in a specific user's profile. In general, even in the case of having data coming from personal players obeying solely to the user's will, we should discard (i) users that do not provide enough data to be processed, and (ii) artists and genres that only appear occasionally. We prefer to sacrifice a big amount of raw data provided those we keep help to identify a few of clearly recurring patterns, even if it is only for a few users, artists or genres.

In order to achieve the above-mentioned cleaning goals we first compute, for each user, the average frequency of each artist/genre in his/her playlist. Then, for each user's dataset, we filter out all those artists/genres for which the playlist length is below the user's overall average playlist length. Finally, in order to get rid of low-frequency playing users, we compute the median value of the number of artists/genres left after the last filtering step, which we will name as "valid" artists/genres. Those users whose number of "valid" artists/genres is below the median percentage value are discarded.

3.3 Prediction and Validation Data Sets

Once we get rid of all the suspected noise, we split our dataset in two groups. One will be used to generate the temporal predictions while the other one will be used to test them. The test set contains all the data in the last year of listening for a given subject. The prediction-generation set contains the data coming from two years of listening previous to the year used in the test set.

3.4 Circular Statistics

Circular statistics are aimed to analyze data on circles where angles have a meaning, which is the case when dealing with daily or weekly cycles. In fact, circular statistics is an alternative to common methods or procedures for identifying cyclic variations or patterns, which include spectral analysis of time-series data or time-domain based strategies [15]. Although these approaches are frequently used, their prerequisites (e.g., interval scaling, regularly spaced data, Gaussianity) are seldom met and, as we mentioned above, these techniques have rarely been used to analyze music-related data and therefore we wanted to explore its potential.

Under the circular statistics framework, variables or data considered to be cyclic in nature are meant to have a period of measurement that is rotationally invariant. In our case this period is referred to the daily hours and the days of the week. Therefore, taking into account the rotationally invariant period of analysis this would be reflected as daily hours that range from 0 to 24, where 24 is considered to be the same as 0. Regarding to the weekly rhythm, Monday at 0h would be considered to be the same as Sunday at 24h.

[1] Last.fm relevance weight of tag *t* to artist *a*, ranging from 0 to 100.

The first step in circular analysis is converting raw data to a common angular scale. We chose the angular scale in radians, and thus we apply the following conversion to our dataset:

$$\alpha = \frac{2\pi x}{k}$$

where x represents raw data in the original scale, α is its angular direction (in radians) and *k* is the total number of steps on the scale where x is measured. In fact, we denote α as a vector of *N* directional observations α_i (i ranging from 1 to *N*). For the daily hour case, x would have values between 0 and 24, and k = 24. Alternatively, for the weekday analysis, x would have a scale from 0 (Monday) to 6 (Sunday) and thus, k = 6. As noted, the effect of this conversion can be easily transformed back to the original scale. Once we have converted our data to angular scale, we compute the *mean direction* (a central tendency measure) by transforming raw data into unit vectors in the two-dimensional plane by

$$r_i = \begin{pmatrix} \cos\alpha_i \\ \sin\alpha_i \end{pmatrix}$$

After this transformation, vectors r_i are vector-averaged by

$$\bar{r} = \frac{1}{N}\sum_i r_i$$

The quantity $\bar{r}$ is the *mean resultant vector* associated to the mean direction, and its length $\bar{R}$ describes the spread of the data around the circle. For events occurring uniformly in time $\bar{R}$ values approach 0 (uniform circular distribution) whereas events concentrated around the mean direction yield values close to 1 (see figure 1 for an example). A null hypothesis (e.g., uniformity) about the distribution of data can be assessed using Rayleigh's [11] or Omnibus (Hodges-Ajne) tests [27], the latter working well for many distribution shapes. Once we have detected significantly modally distributed data by means of both tests, we verify that it wasn't completely pointing to a single day or hour. All the circular statistics analyses presented here have been performed with the CircStat toolbox for Matlab [6].

4. RESULTS

4.1 Data cleaning

As a consequence of the cleaning process, our working dataset now contains data from 466 valid users. The cleaning process has kept 62% of their total playcounts, which corresponds to 4,5% of the initial amount of artists. This dramatic reduction of the artists should not be surprising as many listening records show a "long-tail" distribution, with just a few of frequently played artists, and many of them seldom played. On the other hand, when focusing on musical genre listening, the working dataset includes 515 users, from which 78% of their playcounts has been kept. These playcounts comprise 8.6% of the total number of genres. Again, a long-tail distribution of the amount of listened genres is observed.

4.2 Temporal Patterns of Artist Selection

Once we have cleaned our dataset, we compute the mean circular direction and the mean resultant vector length for each artist and user. Therefore, these values can be considered as a description of the listening tendencies for each artist by each user. Both parameters were calculated for the daily and for the weekly data.

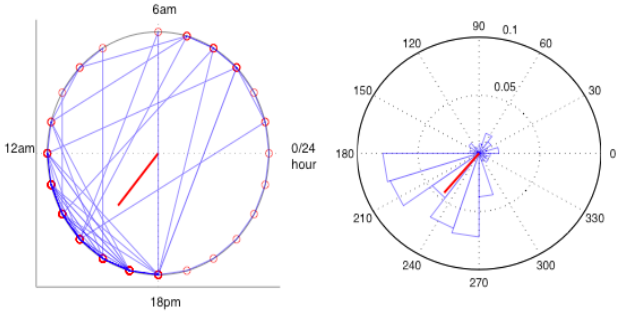


Figure 1. Circular representation of a specific user listening behavior for a specific artist along 24 hours. The left side diagram shows the daily distribution of listening, and the right one the circular histogram. The red line represents the mean vector direction and length in both cases.

In order to assess the relevance of these listening trends, we tested that the distribution of playcounts was different from uniform, and that it was modally distributed (i.e, showing a tendency around an hour or around a day of the week) and discarded those that were not fulfilling these requirements (a null hypothesis rejection probability $p<0.05$ was set for the tests).

In the hour prediction problem, for each listener's clean dataset almost 93% (σ=13) of the artists passed on average the uniformity test (i.e., listening to them is meant to be concentrated around a specific hour). However, considering the raw dataset, only a per-user average of 7% (σ=3.2) of the artists show a listening hour tendency. For the weekly approach, the per-user average in the clean dataset is 99.8% (σ=0.8), indicating that there are some artists showing a clear tendency towards a preferred listening day. Considering the original raw dataset, they correspond to a 7.5% (σ= 3.2) of all the played artists.

Data from 466 users, including 7820 different songs and a grand total of 23669 playcounts were used in the validation of the temporal listening patterns of artists. For each user and artist we computed a "hit" if the absolute difference between the playing day in the prediction and test conditions, expressed as a circular mean value in radians, was less than 0.45 (the equivalent to a half-a-day error). For the time of the day a half-an-hour error was accepted, corresponding to a difference between the predicted and the observed time of less than 0.13 radians.

When predicting the day of listening, an overall 32.4% of hits was found for the songs in the test collection, which exceeds by far the chance expectations (1/7=14.28%). As the final goal of the model is providing user-specific contextual recommendation, an additional per-user analysis yielded 34.5% of hits (σ=17.8). Identical data treatment was done with the time of the day yielding an overall 17.1% of hits (chance expectation baseline: 1/24=4.1%) and a per-user hit rate of 20.5% (σ=16.4).

4.3 Temporal Patterns of Genre Selection

Data from 456 users, including more than 5100 songs and 117 genres, were used for the validation of the genre-related patterns. In order to consider a "hit" in the prediction of listening time and day for a given genre, we set the same thresholds than for evaluating the artist prediction. For the time of the day an overall 22.6% (and per-user 23.2%) of accurate predictions was found. It is interesting to note that relaxing the required accuracy of the prediction to plus/minus one hour error we reached 39.9% of

average hits and per-user average 41% (σ=28.4). For the day of the week, the overall hit percent was 40.9%, while the per-genre average and the per-user average were, respectively, 40.7% (σ=24.1) and 41.7% (σ=26.3). It is interesting to note that among the best predictable genres we find many of infrequent ones but also many of the most frequent ones.

5. CONCLUSIONS

The present study is, as far as we know, the first one inquiring the possibility that our music listening behavior may follow some detectable circadian and ultradian patterns, at least under certain circumstances. We have discovered that a non-negligible amount of listeners tend to prefer to listen to certain artists and genres at specific moments of the day and/or at certain days of the week. We have also observed that, respectively for artists and for genres, 20% and 40% time-contextualized music recommendations can be successful. In our future work agenda, more sophisticated prediction models will be tested, and also ways to implement them into existing music recommenders.

6. ACKNOWLEDGMENTS

Our thanks to Òscar Celma who kindly shared the Last.fm data file, accessible from this URL:
http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-1K.html

7. REFERENCES

[1] Anderson, M., Ball, M., Boley, H., Greene, S., Howse, N., Lemire, D, and McGrath, S. 2003. Racofi: A rule-applying collaborative filtering system. In Proc. of COLA'03.

[2] Baltrunas, L. and Amatriain, X. 2009. Towards Time-Dependant recommendation based on implicit feedback. RecSys09 Workshop on Context-aware Recommender Systems (CARS-2009).

[3] Balzer, H.U. 2009. Chronobiology as a foundation for and an approach to a new understanding of the influence of music. In R. Haas and V. Brandes (Eds.), *Music that Works*. Wien/New York: Springer Verlag.

[4] Barabasi, A.L. 2010. *Bursts: The Hidden Pattern Behind Everything We Do*. New York: Dutton Books.

[5] Beran, J. 2004. *Statistics in Musicology*, Boca Raton: CRC.

[6] Berens P., 2009, CircStat, a Matlab Toolbox for Circular Statistics, *Journal of Statistical Software*, 31, 10.

[7] Boström, F. 2008. AndroMedia - Towards a Context-aware Mobile Music Recommender. Master's thesis, University of Helsinki, Faculty of Science, Department of Computer Science. https://oa.doria.fi/handle/10024/39142.

[8] Coppola, P., Della Mea, V., Di Gaspero, L., Menegon, D., Mischis, D., Mizzaro, S., Scagnetto, I. and Vassena, L. 2009. The context-aware browser. IEEE Intelligent Systems, 25,1, 38-47.

[9] Dressler, K. and Streich, S. 2007. Tuning Frequency Estimation Using Circular Statistics. 8th Int. Conf. on Music Information Retrieval (ISMIR-2007), 357-360.

[10] Eagle, N. and Pentland, A.S. 2009. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63, 7, 1057-1066.

[11] Fisher N.I., 1993, *Statistical Analysis of circular data*, Cambridge: Cambridge University Press.

[12] Foster, R.G., and Kreitzman, L. 2005. *Rhythms of Life: The Biological Clocks that Control the Daily Lives of Every Living Thing*. Yale: Yale University Press.

[13] Hargreaves, D. J. and North, A. C. 1999. *The functions of music in everyday life*: Redefining the social in music psychology. Psychology of Music 27, 71-83.

[14] Koren, Y. 2009. Collaborative filtering with temporal dynamics, New York, NY, USA, 447-456.

[15] Kubiak, T. and Jonas, C. 2007. Applying circular statistics to the analysis of monitoring data: Patterns of social interactions and mood. *European Journal of Personality Assessment*, 23, 227-237.

[16] Larsen, R.J., Augustine, A.A., and Prizmic, Z. 2009. A process approach to emotion and personality: Using time as a facet of data. *Cognition and Emotion*, 23, 7, 1407-1426.

[17] Lee, J.S. and Lee, J.C. 2008. Context awareness by case-based reasoning in a music recommendation system. *4th Int. Conf. on Ubiquitous Computing Systems*, 45-58.

[18] Lloyd, D., and Rossi, E. 2008. *Ultradian Rhythms from Molecules to Mind: a new vision of life*. New York: Springer.

[19] Lombardi, S., Anand, S. and Gorgoglione, M. 2009. Context and Customer Behavior in Recommendation. RecSys09 Workshop on Context-aware Recommender Systems.

[20] Neuhaus, F., 2010. *Cycles in Urban Environments: Investigating Temporal Rhythms*. Saarbrücken: LAP.

[21] Radocy, R.E. and Boyle, J.D. 1988. *Psychological Foundations of Musical Behavior* (2nd ed.) Springfield, IL: Charles C. Thomas.

[22] Rentfrow, P.J. and Gosling, S.D. 2003. The do re mi's of everyday life: The structure and personality correlates of music preferences. *Journal of Personality and Social Psychology*, 84, 6, 1236-1256.

[23] Reynolds, G., Barry, D., Burke, T., and Coyle, E. 2008. Interacting with large music collections: towards the use of environmental metadata. *IEEE International Conference on Multimedia and Expo*, 989-992.

[24] Saarikallio, S., and Erkkilä, J. 2007. The role of music in adolescents' mood regulation. *Psych. of Music*, 35, 1, 88-109.

[25] Su, J.H., Yeh, H.H., Yu, P.S., Tseng, V. 2010. Music recommendation using content and context information mining. *IEEE Intelligent Systems*, 25, 16-26.

[26] Valcheva, M. 2009. Playlistism: a means of identity expression and self-representation. Technical Report, Intermedia, University of Oslo. http://www.intermedia.uio.no/download/attachments/43516460/vit-ass-mariya_valcheva.pdf?version=1

[27] Zar J.H. 1999, *Biostatistical Analysis* (4th edition), Upper Saddle River, NJ: Prentice Hall.

Using Song Social Tags and Topic Models to Describe and Compare Playlists

Ben Fields, Christophe Rhodes and Mark d'Inverno
Department of Computing
Goldsmiths University of London
New Cross
London, SE14 6NW
United Kingdom
[b.fields | c.rhodes | dinverno]@gold.ac.uk

ABSTRACT

Playlists are a natural delivery method for music recommendation and discovery systems. Recommender systems offering playlists must strive to make them relevant and enjoyable. In this paper we survey many current means of generating and evaluating playlists. We present a means of comparing playlists in a reduced dimensional space through the use of aggregated tag clouds and topic models. To evaluate the fitness of this measure, we perform prototypical retrieval tasks on playlists taken from radio station logs gathered from Radio Paradise and Yes.com, using tags from Last.fm with the result showing better than random performance when using the query playlist's station as ground truth, while failing to do so when using time of day as ground truth. We then discuss possible applications for this measurement technique as well as ways it might be improved.

Categories and Subject Descriptors

H.5.5 [**Sound and Music Computing**]: Signal analysis, synthesis, and processing; H.5.1 [**Multimedia Information Systems**]: Evaluation/methodology

Keywords

LDA, Topic Models, playlists, music, similarity, information retrieval, metric space, social tags

1. INTRODUCTION

Inherent to the design of any recommender or retrieval system is a means of display or delivery of selected content. For a system that recommends music this means playback of an audio file. Listening to or playing a piece of music take the length time of that piece of music. Given this link between music and time, when considering what information is relevant for a recommendation it is vital to consider the context of time; that is, what music has been played before or will be played after the current recommended song. Yet little is understood about how playback order affects the success or failure of a recommendation of a piece of music. Whether a system makes user-based, object-based or hybrid recommendations, a better awareness and use of playback order will yield an improved music recommender system.

In order to take advantage of the effect of playback order, it is necessary to have some means of comparing playlists with one another. While ratings-based generic recommender strategies could be employed, such techniques could only be used in systems which allow for the rating of playlists directly (as opposed to the much more common rating of member songs).Alternatively, n distance measure between playlists can be used to facilitate the prediction and generation of well-ordered lists of song sequences for recommendation. This has the advantage being applicable to the vast majority of existing playlist generation systems, many of which do not to collect playlist level ratings from their users. Further, a measure of playlist distance has a number of other applications in music recommender and discovery systems including label propagation, predictive personalization and context tuning to name a few.

In this paper we propose an objective distance measure between playlists. To better understand why such a measure is needed, Section 2 provides background information in existing playlist generation and evaluation techniques. While any sufficiently expressive and low-dimensional feature is compatible with our playlist measure, we use a novel social tag-based feature in this paper. This song-level feature is detailed in Section 3. This is followed by an explanation of our distance measurement itself in Section 4. Putting this into practice, we detail some proof of concept evaluation in Section 5. We discuss the results of this evaluation and possible extensions in Section 6.

2. PLAYLIST AS DELIVERY MECHANISM

In this section we survey the use of playlists in the delivery of content in existing recommendation and retrieval systems. This is followed by a review of current evaluation methods for generated playlists. These two survey points will show both the widespread use of playlist generation in music recommendation and discovery systems and the need for more quality evaluation of these systems.

While this brief survey is focused on automatic playlist generation, there is a wealth of both academic and lay work discussing various aspects manual human-driven playlist con-

WOMRAD 2010 Workshop on Music Recommendation and Discovery, colocated with ACM RecSys 2010 (Barcelona, SPAIN)

struction that may be of interest to the reader. Work in this area tends to deal with radio (e.g. [1]) or club and dance disc jockeys (e.g. [13]), being the two principal areas where the explicit construction of ordered lists of songs are tied to the field. It is with these areas of manual playlist construction in mind that we will examine past efforts in both automatic playlist construction and evaluation techniques.

2.1 Usage in the Wild

There have been many music recommendation and retrieval systems that employ some kind of automatic playlist construction within their system. Frequently this is done as a means of content delivery or, less often, as a way of facilitating human evaluation of an underlying process such as content-based music similarity or recommendation. What follows is a brief survey of existing methods of playlist generation both with and without human intervention.

A web based system for personalized radio is detailed in [20]. In this early system users create and publish playlists facilitated through a process analogous to collaborative filtering. This results in quasi-automatic playlist creation, with any sequence ordering depending entirely on the user. Another variation of the social interaction intermediary is shown in [27], which presents the *Jukola* system. This system creates playlists via democratic vote on every song using mobile devices of listeners in the same physical space. Furthering the ideas of collaborative human generation, [25] shows a system called *Social Playlist*. This system is based on the idea of social interaction through playlist sharing, integrating mobile devices and communal playback.

A fully automatic rule-based system is described in [2]. This system uses existing metadata such as artist name, song title, duration and beats per minute. The system is designed from the ground up to be scalable and is shown to work given a database of 200000 tracks. An approach that is derived from recommender systems is seen in [4]. Here the authors use the ratings and personalization information to derive radio for a group. An attempt to optimize a playlist based on known user preference as encoded in song selection patterns is shown in [30]. This effort uses Gaussian process regression on user preference to infer playlists. The system uses existing a priori metadata as the features for selection. A means of using webmining derived artist similarity with content-based song similarity is used to automatically generate playlists in [22]. This system combined these two spaces in such a way as to minimize the use of signal analysis. A byproduct of this optimization is improved playlist generation as is shown in a small evaluation with human listeners.

The *Poolcasting* system is detailed in [5, 6]. Poolcasting uses dynamic weighting of user preferences within a group of users who are all listening to a common stream with the goal of minimizing displeasure across the entire group. This results in a system that is very similar to popular commercial radio in terms of its output. A method for created playlists using an artist social graph, weighted with acoustic similarity is shown in [17]. This method takes a start and end song and constructs a playlist using maximum flow analysis on the weighted graph. Another technique for playlist construction based on the selection of paths between the start and end songs is shown in [18]. In this system content-based similarity is used to project a set of songs onto a 2-D map, then a path is found from the start song to the end song with the goal of minimizing the step size between each member song. A recent approach uses co-occurrence in n-grams extracted from the internet radio station Radio Paradise[1] to deform a content-based similarity space [26]. This deformed space is then used in a manner that is similar to [18] to generate paths from one song to another, minimizing step distance throughout the path.

Also of note is [31], which in contrast to most of the previous systems, uses nearest neighbor co-occurrence in radio playlist logs to determine song similarity. While the evaluation was preliminary this method shows promise.

2.2 Evaluation Methods

The most prevalent method of evaluation used in playlist generation systems is direct human evaluation by listening. The system detailed in [29], a rule-based automatic playlist generator that uses features derived from metadata, is similar to [2, 30]. Of note in [29] is the thorough human listener testing which shows the automatic playlist generator performing considerably better than songs ordered randomly. This evaluation, though better than most, still fails to compare the automatic playlists against human expert playlists. Additionally, to reduce test time, the evaluation uses arbitrary one minute clips from the songs rather than the entirety of the song or an intentionally chosen segment. A content-based similarity playlist generator with a novel evaluation is seen in [28]. Here the authors track the number times the user presses the *skip* button to move on from the currently playing song. All songs that are skipped are considered *false positives* and those that are completely played are treated as *true positives*. From this many standard information retrieval techniques can be used in the evaluation, resulting in a rich understanding of the results. Ultimately, it is still human user listening evaluation though and its biggest drawback is playback time. Assuming an average song length of five minutes it would take an an hour and 40 minutes (per listener) to listen to 20 songs with no time for the skipped songs. This skip-based evaluation framework is further used in [12] where existing last.fm user logs (which include skip behavior) are analyzed using fuzzy set theory to determine playlist generation heuristics in the system. Additionally, many systems of playlist generation lack formal evaluation all together.

2.3 Summary

While a number of techniques have been employed to create playlists for a variety of functions, there exist limited techniques in the evaluation of generated playlists. These evaluation techniques rely heavily on time consuming human evaluation. Beyond that, there is no studied means to objectively compare one playlist with another. In Section 4 we will propose just such a means. First we will describe a novel song level feature based on tags. A tag-based feature will encode socio-cultural data that is missing from analogous content-based features, though social tags bring about some other problems.

3. TOPIC-MODELED TAG-CLOUDS

In order to encode playlists in a low dimensional representation we must first represent their member songs in as a low dimensional vector. Here we use a Topic-Modeled Tag Cloud (TMTC) as a pseudo-content-based feature, in a way

[1] `http://radioparadise.com`

Figure 1: The tag cloud for Bohemian Crapsody by Sickboy, from Last.fm.

that is functionally analogous to various pure content-based methods. Using tags and topic models in this way is novel and what follows is an explanation of the process of building this feature.

3.1 Tags as Representation

A *tag* is a word or phrase used to describe a document of some kind, typically on the Web. Various kinds of documents are described using tags on the Web including photos[2], videos[3] and music[4]. An aggregated collection of tags, weighted by the number of users who ascribe it to a given object, is commonly referred to as a *tag cloud*.

Tag clouds get their name from the most common visualization method used with them, where each tag is displayed with the font size in proportion to the weight, arranged in a way that resembles a cloud. An example of a tag cloud[5] can be seen in Figure 1 As can be seen in this example, tag clouds provide a rich description of the music it describes. Tags and collections of tags in various forms provide the basis for many techniques within music informatics including recommendation, retrieval and discovery applications [3,23].

In addition to human generated tags being used, there is some research directed toward the automatic application of tags and inference of associated weights on unlabeled pieces of music [7,9,16,21].

3.2 Reducing the Dimensionality

There exist some techniques (such as [8]) to determine semantic clustering within a tag cloud; however, these systems are built to facilitate browsing and do not create a sufficiently reduced dimensional representation. The previous work of [24] comes the closest to the needed dimensional reduction, also dealing with social tags for music. This work, through the use of aspect models and latent semantic analysis, brings the dimensionality down into the hundreds, while preserving meaning. This order of dimensions is still too high to compute meaningful distance across multi-song playlists. A feature with dimensionally of the order 10^2 would suffer from the curse of dimensionality [33]: because of its high dimensionality, any attempt to measure distance becomes dominated by noise. However, a technique developed for improved modelling in text information retrieval, *topic models* provide the reduced dimensional representation we require. Topic models are described in [10] as "probabilistic models for uncovering the underlying semantic structure of [a] document collection based on a hierarchical Bayesian analysis of the original text." In topic modeling, a *document* is transformed into a *bag of words*, in which all of the words of a document are collected and the frequency of the occurrence in recorded. We can use the weighted collection of tags in a tag cloud as this bag of words, with tags serving as tokenized words.

[2]e.g. `http://flickr.com`

[3]e.g. `http://youtube.com`

[4]e.g. `http://last.fm` or `http://musicbrainz.org`

[5]This tag cloud is for the track Bohemian Crapsody by the artist Sickboy. The tags and the rendering both come from last.fm, available at `http://www.last.fm/music/Sickboy/_/Bohemian+Crapsody/+tags`

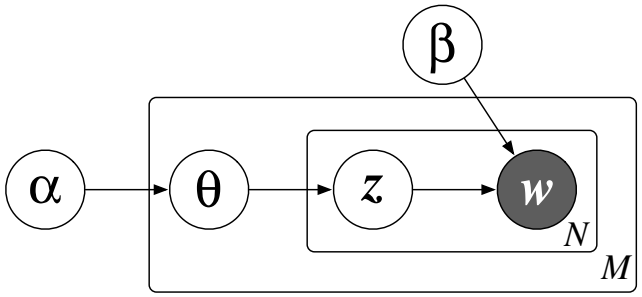


Figure 2: The graphic model of LDA [11]. The replicates are represented as the two boxes. The outer box M represents the corpus of documents, while the inner box N represents the repeating choice of topics and words which make up each document.

There are a few different ways of generating topic models; for our feature generation we will be using latent Dirichlet allocation [11], treating each tag cloud as a bag-of-words. In LDA, documents (in our case tags clouds of songs) are represented as a mixture of implied (or *latent*) topics, where each topic can be described as a distribution of words (or here, tags).More formally give the hyper-parameter α, and the conditional multinomial parameter β, Equation 3.2 gives the joint topic distribution θ, a set of N topics $\mathbf{z}$ and a set of N tags $\mathbf{w}$.

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta) \qquad (1)$$

In Figure 2 LDA is shown as a probabilistic graphical model. In order to create topic models using LDA, we need to specify $p(\theta|\alpha)$ and $p(z_n|\theta)$. We estimate our parameters empirically from a given corpus of tag clouds. This estimation is done using *variational EM* as described in [11].This allows topic distributions to be generated in an unsupervised fashion, though the number of topics in a corpus must be specified a priori.

Once the LDA model is generated, it is used to infer the

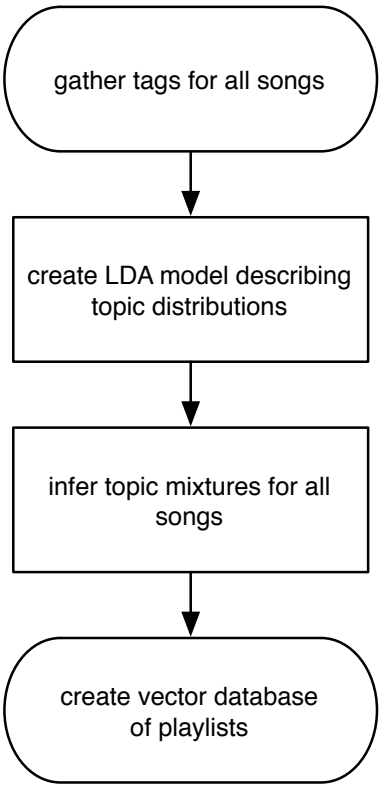


Figure 3: The complete process for construction of a TCTM feature set.

mixture of topics present in the tag cloud for a given song. This is done via *variational inference* which is shown in [11] to estimate the topic mixture of a document by iteratively minimizing the KL divergence from variational distribution of the latent variables and the true posterior $p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta)$.

This process in it's entirety is shown as a block diagram in Figure 3. Once this process is completed for every song in our dataset, we will have a single vector with a dimensionality equal to the number of topics in our LDA whose entries indicate topic occupancy for that song.

4. PLAYLISTS AS A SEQUENCE OF TOPIC WEIGHTS

Given the single vector per song reduction, we represent the playlists these song are in as ordered sequences of these vectors. Thus each playlist is represented as a $l \times d$-dimensional vector, where l is the number of songs in a given playlist and d is the number of topics in our LDA model.

4.1 Measuring Distance

To both manage and measure the distance between these $l_i \times d$ dimensional vectors we use audioDB[6]. The use of audioDB to match vectors of this type is detailed in [32]. Briefly, distance is calculated by means of a multidimensional Euclidian measure. Here l_i is an arbitrary length subsequence of i vectors. In practice, i is Casey:2008selected to be less than or equal to the smallest sequence length for a complete playlist in a dataset. The distance between two playlists is then the minimum distance between any two length i sub-vectors drawn from each playlist. One effect of this technique is easy handling of playlists of unequal length.

This type of distance measurement has been used with success on sequences of audio frames [14, 15]. The distance measure in use between vectors can also be changed. In particular there has been work showing that statistical features (such as topic models) may benefit from the use of Manhattan distance [19], however for our prototypical evaluation we have used simple Euclidean distance as seen in equation **??** above.

5. EVALUATION

The goal of our evaluation is to show the fitness of our distance measurement through preliminary retrieval tests: searching for playlists that start at the same time of day as our query playlist and searching for the playlists from the same station from a database of stations of the same genre. We examine the logs of a large collection of radio stations, exhaustively searching example sets. Through precision and recall we see that our measure organizes playlists in a predictable and expected way.

5.1 Dataset

In order to test these proposed techniques a collection of radio station logs were gathered. These logs come from a collection of broadcast and online stations gathered via Yes.com[7]. The logs cover the songs played by all indexed stations between 19-26 March 2010. For our evaluation task using this data source we looked at subsets of this complete capture, based on genre labels applied to these stations. Specifically we examine stations of the genres *rock* and *jazz*. The complete Yes.com dataset also includes stations in the following genre categories: *Christian*, *Country*, *Electronica*, *Hip-Hop*, *Latin*, *Metal*, *Pop*, *Punk*, *R&B/Soul*, *Smooth Jazz* and *World*. These labels are applied by the stations themselves and the categories are curated by Yes.com. Additionally, the play logs from Radio Paradise[8] from 1 January 2007 to 28 August 2008 form a second set. We then attempted to retrieve tag clouds from Last.fm[9] for all songs in these logs. When tags were not found the song and its associated playlist were removed from our dataset

These logs are then parsed into playlists. For the radio logs retrieved via the Yes api, the top of every hour was used as a segmentation point as a facsimile for the boundary between distinct programs. This is done under the assumption that program are more likely than not to start and finish on the hour in US commercial broadcast. Note that this method of boundary placement will almost certainly oversegment radio programs as many radio programs are longer than one hour. However, given that our distance measure compares fixed length song sequences across playlists, this over-segmentation should produce only minimal distortion in our results. The Radio Paradise logs include all the *links* or breaks between songs where the presenter speaks briefly. For experiments using the Radio Paradise logs these links are used as playlist boundaries. This leads to a slight difference in the type of playlist used from Radio Paradise versus Yes.

[6]source and binary available at `http://omras2.doc.gold.ac.uk/software/audiodb/`

[7]`http://api.yes.com`

[8]`http://www.radioparadise.com/`

[9]`http://last.fm`

source	S_t	S_{mt}	P_t	$P_{avg(time)}$	$P_{avg(songs)}$
whole set	885810	2543	70190	55min	12.62
"Rock" stations	105952	865	9414	53min	11.25
"Jazz" stations	36593	1092	3787	55min	9.66
"Radio Paradise"	195691	2246	45284	16min	4.32

Table 1: Basic statistics for both the radio log datasets. Symbols are as follows: S_t is the total number of song entries found in the dataset; S_{mt} is the total number of songs in S_t where tags could not be found; P_t is total number of playlists; $P_{avg(time)}$ is the average runtime of these playlists and $P_{avg(songs)}$ is the mean number of songs per playlist.

The playlists coming from Radio Paradise represent strings of continuously played songs, with no breaks between the songs in the playlists. The playlists from Yes are approximations of a complete radio program and can therefore contain some material inserted between songs (e.g. presenter link, commercials).

Statistics for our dataset can be see in Table 1 we then use the tags clouds for these songs to estimate LDA topic models as described in Section 3[10]. For all our experiments we specify 10 topic models a priori. The five most relevant tags in each of the topics in models trained on both the rock and jazz stations can be seen Table 2.

5.2 Daily Patterns

Our first evaluation looks at the difference between the time of day a given query playlist starts and the start time for the closest n playlists by our measure. For this evaluation we looked at the 18 month log from Radio Paradise as well as the "Rock" and "jazz" labelled stations from Yes.com, each in turn. Further we used a twelve hour clock to account for The basis for this test relies on the hypothesis that for much commercial radio content in the United States, branding of programs is based on daily repeatable of tone and content for a given time of day. It should therefore be expected that playlists with similar contours would occur at similar times of day across stations competing for similar markets of listeners.

Figure 4 shows the mean across all query playlists of the time difference for each result position for the closest n results, where n is 200 for the Radio Paradise set and 100 for the Yes.com set. The mean time difference across all three sets is basically flat, with an average time difference of just below 11000 or about three hours. Given the maximum difference of 12 hours, this result is entirely the opposite of compelling, with the retrieved results showing no corespondance to time of day. Further investigation is required to determine whether this is a failure of the distance metric or simply an accurate portrail of the radio stations logs. A deeper examination of some of the Yes.com data shows some evidence of the latter case. Many of the playlist queries exactly match (distance of 0) with the entirity of the 200 returned results. Further these exact match playlists are repeated evenly throughout the day. One of these queries is shown in Figure 5. The existance of these repeating playlists throughout the day, ensures this task will not confirm our hypothesis, perhaps due to progaming with no reliance on time of day, at least in the case of Radio Paradise.

[10]Our topic models are created using the open source implementation of LDA found in the gensim python package available at `http://nlp.fi.muni.cz/projekty/gensim/` which in turn is based on Blei's C implementation available at `http://www.cs.princeton.edu/~blei/lda-c/`

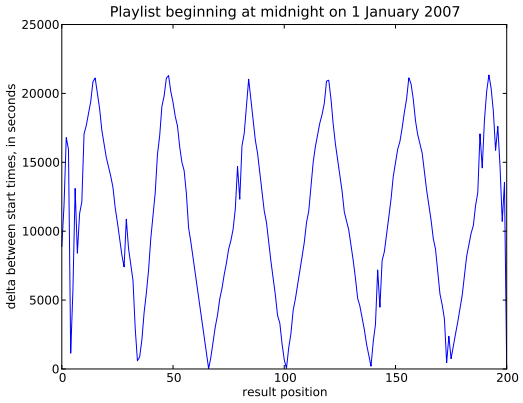


Figure 5: The time of day difference from the query playlist for 200 returned results, showing even time of day spread. Note that all the results show here have a distance of 0 from the query.

5.3 Inter-station vs. Intra-station

In this evaluation we examined the precision and recall of retrieving playlists from the same station as the query playlist. Here we looked at the "Rock" and "Jazz" labelled stations retrieved via the Yes API, each in turn. Similar to the first task, it is expected that a given station will have its own *tone* or particular *feel* that should lead to playlists from that station being more apt to match playlist from their generating station then with other stations from the same genre. More formally, for each query we treat returned playlists as relevant, true positives when they come from the same station as the query playlist and false positives otherwise. Based on this relevance assumption, precision and recall can be calculated using the following standard equations.

$$P = \frac{|\{relevantplaylists\} \bigcap \{retrievedplaylists\}|}{|\{retrievedplaylists\}|} \quad (2)$$

$$R = \frac{|\{relevantplaylists\} \bigcap \{retrievedplaylists\}|}{|\{relevantplaylists\}|} \quad (3)$$

The precision versus recall for a selection of stations' playlists from both the "Rock" and "Jazz" stations are show in Figure 6. When considering the precision and recall performance it

station label	t_1	t_2	t_3	t_4	t_5
"Rock"	Snow Patrol	Bob Marley	female vocalists	aupa Pete	80s
	rumba	Feist	Anna Nalick	whistling	new wave
	90s	john mayer	Chicas	Triple J Hottest 100	david bowie
	green day	drunk love	playlist 2009	review	neuentd
	Dynamit	feist backing vocals	Sarah McLachlan	fun as fuck	synth pop
"Jazz"	motown	john mayer	60s	Sade	Flamenco
	soul	acoustic	jazz - sax	deserves another listen	tactile smooth jazz
	70s	corinne bailey rae	acid jazz	till you come to me	guitar ponder
	funk	bonnie raitt	reggae	piano	cafe mocha
	Disco	David Pack 2	cool jazz	2010	wine

station label	t_6	t_7	t_8	t_9	t_{10}
"Rock"	classic rock	TRB	reminds me of winter	Needtobreathe	Krista Brickbauer
	60s	ElectronicaDance	kings of leon	plvaronaswow2009	day end
	70s	mysterious	songs that save my life	The Script	i bought a toothbrush
	The Beatles	best songs of 2009	songs to travel	brilliant music	bluegrass
	the rolling stones	tribute to george	Muse	van morrison	omg
"Jazz"	follow-up	rnb	female vocalists	classic rock	Smooth Jazz
	jazz	soul	norah jones	80s	saxophone
	instrumental	female vocalists	dido	rock	smooth jazz sax
	guitar	Neo-Soul	jazz	70s	contemporary jazz
	latin jazz	Robin Thicke	vocal jazz	yacht rock	instrumental

Table 2: The five most relevant tags in each topic. Upper model is all the Yes.com Rock stations, lower model is all Yes.com Jazz stations.

is useful to compare against random chance retrieval. There are 100 stations labeled "Rock" and 48 labeled "Jazz". Under chance retrieval a precision of 0.01 would be seen for "Rock" and 0.0208 for "Jazz".

5.4 Summary

Two different evaluation tasks have been run using real world radio log data to examine the usefulness of our playlist match technique. The first of these, an examination the time difference was flat across result length variance. While this implies lack of discrimination into daily patterns, it is not possible to determine from the available data whether this is an accurate reflection of the progamming within the dataset or distance measure not being sufficient for the task. The second task shows the performance of retrieving hourly playlists from a selection of stations using playlists from that station as a query. Here we see a great deal of promise, especially when comparing the query results against random chance, which it outperforms considerably.

6. CONCLUSIONS

Having reviewed recent work in various methods of playlist generation and evaluation in Section 2, it is apparent that there is a need for better ways to objectively compare playlists to one another. We detailed a method of doing so in Section 4, though first, to better filter content-based data through listeners' experience we presented a novel tag-based feature, TMTC, using tags summarized using LDA topic models in Section 3. This was follow by two task evaluations to examine out playlist matching technique and song feature on real world playlist data from radio logs in Section 5.

While our evaluation shows the promise of this technique on sampled data, there is much room for improvement. Principal among these is the exploration of non-Euclidean distance measures. Manhattan distance (or L_1) seems to have the most direct applicability and its use could prove to be quite beneficial. Another area for future work is in the use of the measure on further data and datasets. One of the best ways to improve here would be in the use of datasets with a more exact known ground truth, in order to best apply known recommender and retrieval evaluation methods to them.

This leads to a further avenue of future work, testing the measure against direct human evaluation. While our matching technique has many uses with recommendation and discovery, if it proved to align with human evaluation it would be considerably more useful.

7. ACKNOWLEDGMENTS

This work is supported in part by the Engineering and Physical Sciences Research Council via the Online Music Recognition And Searching II (OMRAS2) project, reference number EP/E02274X/1. Additional support provided as part of the Networked Environments for Music Analysis (NEMA) project, funded by The Andrew W. Mellon Foundation. Thanks also to Paul Lamere for some dataset acquisition assistance.

8. REFERENCES

[1] J. A. Ahlkvist and R. Faulkner. 'Will This Record Work for Us?': Managing Music Formats in Commercial Radio. *Qualitative Sociology*, 25(2):189–215, June 2002.

[2] J.-J. Aucouturier and F. Pachet. Scaling up playlist generation. In *Proc. IEEE International Conference*

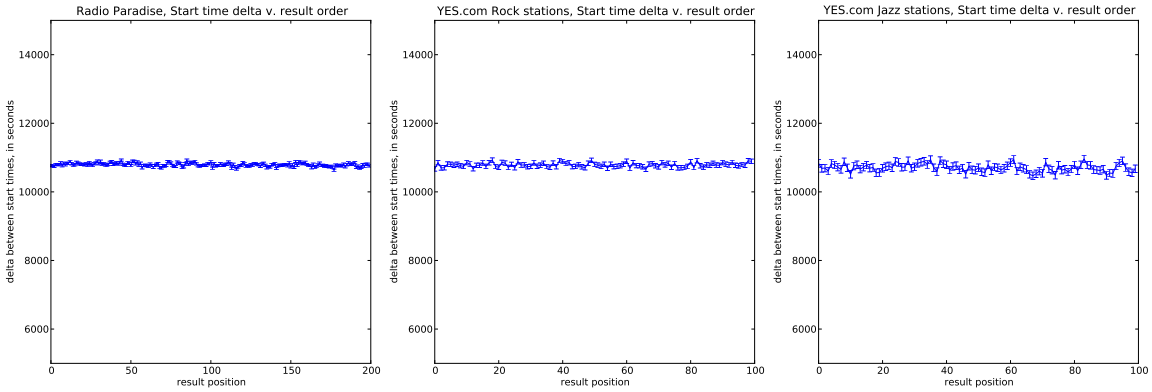


Figure 4: The mean start time difference, with squared error of the mean.

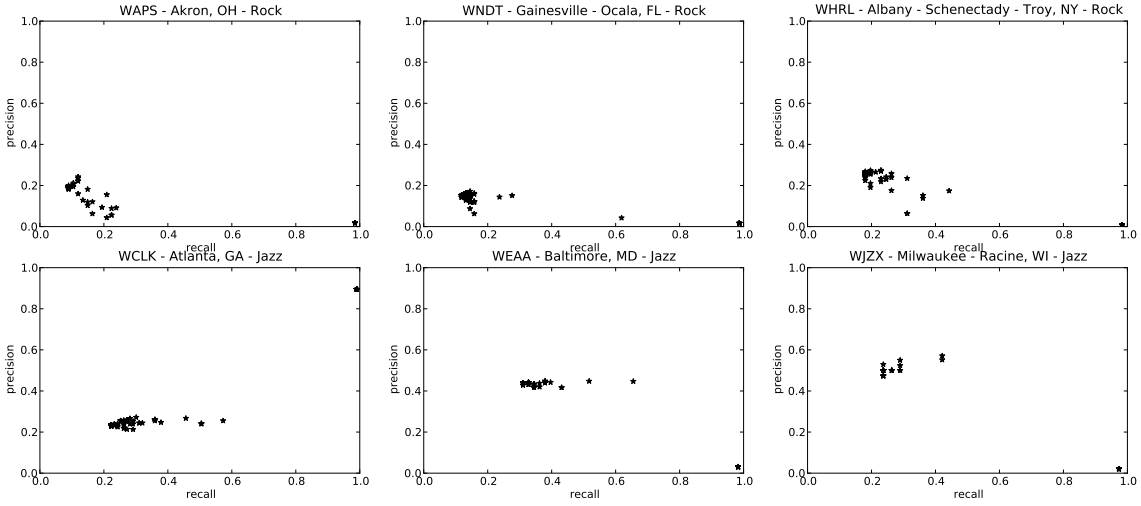


Figure 6: Precision versus Recall for six stations when using their hourly playlists to query for other playlists from the same station. In each query the number of results retrieved is selected to maximize the F1 score.

on Multimedia and Expo, 2002.

[3] J.-J. Aucouturier and E. Pampalk. Introduction-from genres to tags: A little epistemology of music information retrieval research. *Journal of New Music Research*, 37(2):87–92, 2008.

[4] P. Avesani, P. Massa, M. Nori, and A. Susi. Collaborative radio community. In *AH '02: Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 462–465. Springer Berlin / Heideberg, January 2002.

[5] C. Baccigalupo. *Poolcasting: an intelligent technique to customise music programmes for their audience.* PhD thesis, Institut d'Investigació en Intelligència Artificial, 2009.

[6] C. Baccigalupo and E. Plaza. Sharing and combining listening experience: A social approach to web radio. In *Proc. of the International Computer Music Confernce*, 2007.

[7] L. Barrington, D. Turnbull, and G. Lanckriet. Auto-tagging music content with semantic multinomials. In *Proc. of Int. Conference on Music Information Retrieval*, 2008.

[8] G. Begelman, P. Keller, and F. Smadja. Automated Tag Clustering: Improving search and exploration in the tag space. In *In Proc. of the Collaborative Web Tagging Workshop at WWW'06*, 2006.

[9] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere. Autotagger: a model for predicting social tags from acoustic features on large music databases. *Journal of New Music Research*, 37(2):101–121, 2008.

[10] D. Blei and J. Lafferty. *Topic Models.* Text Mining: Theory and Applications. Taylor and Francis, 2009.

[11] D. M. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[12] K. Bosteels, E. Pampalk, and E. E. Kerre. Evaluating and analysing dynamic playlist generation heuristics

using radio logs and fuzzy set theory. In *Proc. of Int. Conference on Music Information Retrieval*, October 2009.

[13] B. Brewster and F. Broughton. *Last Night A DJ Saved My Life; The history of the disc jockey*. Headline Book Publishing, London, United Kingdom, 2nd edition, 2006.

[14] M. Casey, C. Rhodes, and M. Slaney. Analysis of minimum distances in high-dimensional musical spaces. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(5):1015 –1028, jul. 2008.

[15] M. Casey and M. Slaney. The importance of sequences in music similarity. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, Toulouse, France, 2006.

[16] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. In *Neural Information Processing Systems Conference (NIPS) 20*, 2007.

[17] B. Fields, K. Jacobson, C. Rhodes, and M. Casey. Social playlists and bottleneck measurements : Exploiting musician social graphs using content-based dissimilarity and pairwise maximum flow values. In *Proc. of Int. Symposium on Music Information Retrieval*, September 2008.

[18] A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer. Playlist generation using start and end songs. In *Proc. of Int. Symposium on Music Information Retrieval*, October 2008.

[19] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover's distance. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[20] C. Hayes and P. Cunningham. Smart radio: Building music radio on the fly. In *Expert Systems 2000*, pages 2–6. ACM Press, 2000.

[21] M. D. Hoffman, D. M. Blei, and P. R. Cook. Easy as cba: a simple probabilistic model for tagging music. In *Proc. of Int. Conference on Music Information Retrieval*, 2009.

[22] P. Knees, T. Pohle, M. Schedl, and G. Widmer. Combining audio-based similarity with web-based data to accelerate automatic music playlist generation. In *Proc. 8th ACM international workshop on Multimedia information retrieval*, pages 147 – 154, 2006.

[23] P. Lamere. Social tagging and music information retrieval. *Journal of New Music Research*, 37(2), June 2008.

[24] M. Levy and M. Sandler. Learning latent semantic models for music from social tags. *Journal of New Music Research*, 37(2):137 – 150, 2008.

[25] K. Liu and R. A. Reimer. Social playlist: enabling touch points and enriching ongoing relationships through collaborative mobile music listening. In *MobileHCI '08: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 403–406, New York, NY, USA, 2008. ACM.

[26] F. Maillet, D. Eck, G. Desjardins, and P. Lamere. Steerable playlist generation by learning song similarity from radio station playlists. In *Proc. of Int. Conference on Music Information Retrieval*, October 2009.

[27] K. O'Hara, M. Lipson, M. Jansen, A. Unger, H. Jeffries, and P. Macer. Jukola: democratic music choice in a public space. In *DIS '04: Proceedings of the 5th conference on Designing interactive systems*, pages 145–154, New York, NY, USA, 2004. ACM.

[28] E. Pampalk, T. Pohle, and G. Widmer. Dynamic playlist generation based on skipping behavior. In *Proc. of Int. Symposium on Music Information Retrieval*, 2005.

[29] S. Pauws and B. Eggen. Pats: Realization and user evaluation of an automatic playlist generator. In *Proc. of Int. Conference on Music Information Retrieval*, 2002.

[30] J. C. Platt, C. J. Burges, S. Swenson, C. Weare, and A. Zheng. Learning a gaussian process prior for automatically generating music playlists. In *Proc. Advances in Neural Information Processing Systems*, volume 14, pages 1425–1432, 2002.

[31] R. Ragno, C. Burges, and C. Herley. Inferring similarity between music objects with application to playlist generation. In *Proc. 7th ACM SIGMM international workshop on Multimedia information retrieval*, 2005.

[32] C. Rhodes, T. Crawford, M. Casey, and M. d'Inverno. Investigating music collections at different scales with audiodb. *Journal of New Music Research, to appear*, 2010.

[33] R. Weber, H. J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proc. of the Intl. Conf. on Very Large Databases*, 1998.

Piloted Search and Recommendation with Social Tag Cloud-Based Navigation

Cédric Mesnage
Faculty of Informatics
University of Lugano
cedric.mesnage@usi.ch

Mark Carman
Faculty of Informatics
University of Lugano
mark.carman@usi.ch

ABSTRACT

We 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. We propose three tag cloud generation models based on popularity, topics and social structure. We 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. We 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. We find statistically significant evidence at 5% confidence level that the topic and social models outperform the popular model.

1. INTRODUCTION

We investigate mechanisms to explore social network information. Our current focus is to use contextual tag clouds as a mean to navigate through the data and control a recommendation system.

Figure 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 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.

This paper is structured as follows. We first discuss related work in the area of tag cloud-based navigation. We then detail models for generating context-aware tag clouds

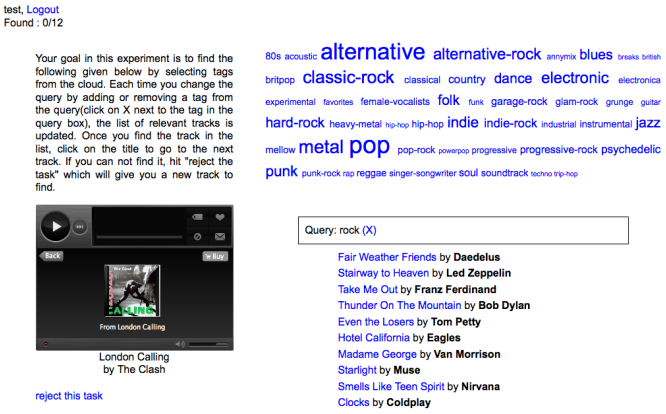


Figure 1: Searching task.

using both social network and topic modeling based approaches, that we have implemented in our prototype tag cloud-based navigation system. We 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.

2. RELATED WORK

2.1 Social Tagging and its Motivations

Research in social tagging is relatively recent with the first tagging applications appearing in the late nineties [12]. The system called Webtagger relied on a proxy to enable users to share bookmarks and assign tags to them. The approach was novel compared to storing bookmarks in the browser's folder in the sense that bookmarks were shared and belonged to multiple categories (instead of being placed in a single folder). The creators argued that hierarchical browsing was tedious and frustrating when information is nested several layers deep.

By 2004, social tagging had reached a point where it was becoming more and more popular, initially on bookmarking sites like Delicious and then later on social media sharing sites such as Flickr and Youtube. Research in social tagging started with Hammond [7] who gave an overview of social bookmarking tools and was continued by Golder *et al.* [5] who provided the first analysis of tagging as a process using tag data from Delicious. They showed that tag data follows a power law distribution, gave a taxonomy of tagging incentives, and looked at the convergence of tag descrip-

WOMRAD 2010 Workshop on Music Recommendation and Discovery, colocated with ACM RecSys 2010 (Barcelona, SPAIN)

Figure 2: Controlled recommendation task.

tions over time for resources on Delicious. The paper lead to the first workshop on tagging [21], where papers mainly discussed tagging incentives, tagging applications (in museums and enterprises), tag recommendation and knowledge extraction. Following this workshop, research in tagging has spread in various already established areas namely in Web search, social dynamics, the Semantic Web, information retrieval, human computer interaction and data mining.

Sen *et al.* [19] 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 (used to recommend tags). Their study focuses on the MovieLens system that consists of user reviews for movies. They categorize tags into three categories: factual, subjective and personal. They then divided users of the system into four groups each with a different user interface: the unshared group didn't see any community tags; the shared group saw random tags from their group; the popular group saw the most popular tags; and the recommendation group used a recommendation algorithm (that selected tags most commonly applied to the target movie and to similar movies). They find that habit and investment influence the users' tag applications, while the community influences a user's personal vocabulary. The shared group produced more subjective tags, while the popular and recommendation group produced more factual tags. The authors also conducted a user survey in which they asked users whether they thought tagging was useful for different tasks: self-expression (50%), organizing (44%), learning (23%), finding (27%), and decision support (21%).

Marlow *et al.* [14, 15] define a taxonomy of design aspects of tagging systems that influence the content and usefulness of tags, namely tagging rights (who can tag), tagging support (suggestion algorithms), aggregation model (bag or set), resource type (web pages, images, etc.), source of content (participants, Web, etc.), resource connectivity (linked or not), and social connectivity (linked or not). They also propose aspects of user incentives expressing the different motivations for tagging: future retrieval, contribution and sharing, attracting attention, playing and competition, self presentation, opinion expression.

Cattuto *et al.* [2, 1] perform an empirical study of tag data from Delicious and find that the distribution of tags over time follows a power law distribution. More specifically they find that the frequency of tags obeys a Zipf's law which is characteristic of self-organized communication systems and is commonly observed in natural language data. They reproduced the phenomenon using a stochastic model, leading to a model of user behavior in collaborative tagging systems.

2.2 Browsing with Tags

Fokker *et al.* [4] present a tool to navigate Wikipedia using tag clouds. Their approach enables the user to select different views on the tag cloud, such as recent tags, popular tags, personal tags or friends tags. They display related tags when the user "mouses over" a tag in the cloud. They do not, however, generate new contextually relevant tag clouds when the user clicks on a tag.

In [16], Millen *et al.* investigate browsing behavior in their Dogear social bookmarking application. The application allows users to browse other peoples' bookmark collections by clicking on their username. They find that most browsing activity of the web site is done through exploring peoples' bookmarks and then tags. They compare the 10 most browsed tags with 10 most used tags applied and find that there is a strong correlation. While their findings do not show that tagging improves social navigation in general, they do show that browsing tags helps users to navigate the bookmark collections of others. Following on from this, Ishikawa *et al.* [10] studied the navigation efficiency when browsing other users' bookmarks. The idea is to decide which user to browse first in order to discover faster the desired information. While relevant to tag-based navigation, this study does not deal with the problem of how best to rank tags in order to improve cloud-based navigation in general.

In [13], Li *et al.* propose various algorithms 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 tags based on time. As discussed by Keller *et al.* [12] a single taxonomy is not necessarily the best way to navigate a corpus, however.

A more comprehensive study was performed by Sinclair *et al.* [20] to examine the usefulness of tag clouds for information seeking. They asked participants to perform information seeking tasks on a folksonomy like dataset, providing 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. The tag cloud displayed 70 terms in alphabetical order with varying font size proportional to the log of its frequency. The authors give the following equation for the font size:

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

where 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 in the cloud brings the user to a

new page listing articles annotated with that tag and a new tag cloud of co-occurring tags. Clicking again on a tag restricts the list to the articles tagged with both tags and so on. The search is based on a TF-IDF 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 performed better when people are asked general questions, for information-seeking, people preferred to use the search box. They conclude that the tag cloud is better for browsing, enhancing serendipity. The participants commented that the search box allows for more specific queries. While similar to our study on tag cloud-based navigation, the work of Sinclair *et al.* [20] differs in a number of important ways: (i) Their aim was to compare tag-based navigation directly with search, while ours is to compare different tag cloud generation methods, based on social network information and topic modeling techniques. (ii) In their study the folksonomy was generated by the participants and is quite small as result, while we rely on an external folksonomy for which scaling becomes an important issue.

In [8], Hassan-Montero *et al.* propose an improvement to tag clouds by ordering the tags according to similarity rather than alphabetically. They use the Jaccard coefficient to measure similarity between tags, which is the ratio between the number of resources in which the two tags both occur and the number in which either one occurs. If $D(w)$ denotes all resources (documents) annotated with tag (word) w, then the similarity is given by:

$$RC(w_1, w_2) = \frac{|D(w_1) \cap D(w_2)|}{|D(w_1) \cup D(w_2)|} \quad (2)$$

The authors then define an additional metric to select which tags to display in each cloud (so as to maximize the number of resources "covered by the cloud"). Their method provided, however, little improvement on the coverage of the selected tags. The tag cloud layout is based on the similarity coefficient. The authors also do not provide a user evaluation of the tag cloud generated.

Kaser *et al.* [11] propose a different algorithm for displaying tag clouds. Their methods concern how to produce HTML in various situations. They also give an algorithm to display tags in nested tables. They do not provide an evaluation regarding the usefulness of the new visual representations.

In [18], Sen *et al.* investigate the 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 movie reviews system, adding to the interface different mechanisms for users to rate the quality of tags. 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 users' ratings. They find that tag selection methods that normalize by user, such as the numbers of users who applied a tag, perform the best.

In [9], Heymann *et al.* investigate the social tag prediction problem, the purpose of which is to predict future tags for a particular resource. The ability to predict tag applications can lead to various enhancements, such as increased recall, inter-user agreement, tag disambiguation, bootstrapping and system suggestion. They collected tag data from Delicious and fetched the web pages for each bookmark. They analyze two methods: The first applies only when the bookmarked items are web pages (and not images, songs, videos, etc.). They develop an entropy based metric which measures how much a tag is predictable. They then extract association rules based on tag co-occurrence 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 this extra information 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.

Ramage *et al.* [17] 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, that "the associations between documents convey information about the relevance of documents to requests". The document clusters are used to solve the problem of query ambiguity by including different clusters in search results.

All of the above mentioned work differs from our current study of tag cloud-based navigation in the following ways: (i) Previous studies have investigated the usefulness of tag clouds primarily from the basic visualization rather than the navigation standpoint. (ii) Those studies explicitly investigating tag cloud based navigation, have concentrated on simple algorithms for generating tag clouds. (iii) Previous studies investigating more sophisticated algorithms for tag prediction have evaluated those algorithms by assessing prediction accuracy on held-out data rather than "in situ" evaluation with real users for a particular application (tag cloud based navigation).

3. 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.

3.1 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.

3.1.1 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:

$$p(w) = \frac{\sum_{d \in D} N_{w,d}}{\sum_{d \in D} N_d} \quad (3)$$

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:

$$p(w|Q) = \frac{\sum_{d \in D(Q)} N_{w,d}}{\sum_{d \in D(Q)} N_d} \quad (4)$$

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.

3.1.2 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:

$$p(w) = \sum_{u \in U} \sum_{u' \in f(u)} \frac{N_{w,u'}}{\sum_{w \in W} N_{w,u'}} \quad (5)$$

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:

$$p(w|w') = \sum_{u \in U} p(w, u|w') \quad (6)$$

$$= \sum_{u \in U} p(w|u) \frac{p(w'|u)p(u)}{p(w')} \quad (7)$$

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:

$$p(w|u) = \frac{\sum_{u' \in f(u)} N_{w,u'}}{\sum_{u' \in f(u)} N_{u'}} \quad (8)$$

Note that since the summation in Equation 7 over all users involves a very large computation, we perform the summation only over the top 200 users as ranked according to the frequency $p(w|u)$.

3.1.3 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 [6]. 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) [6] 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:

$$p(w) = \sum_{z \in Z} p(w|z)p(z) \quad (9)$$

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):

$$p(w|w') = \sum_{z \in Z} p(w|z)p(z|w') \quad (10)$$

$$= \sum_{z \in Z} \frac{p(w|z)p(w'|z)p(z)}{p(w')} \quad (11)$$

3.2 Ranking Resources

We follow a standard Language Modeling [3] 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.

$$p(d|Q) = \frac{p(Q|d)p(d)}{p(Q)} \quad (12)$$

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

$$score(d|Q) = p(Q|d)p(d) \quad (13)$$

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 $p(w|d)$:

$$score(d|Q) = p(Q|d)p(d) \approx p(d) \prod_{w \in Q} p(w|d) \quad (14)$$

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 :

$$score(d|Q) =_{ranking} \log p(d) + \sum_{w \in Q} \log p(w|d) \quad (15)$$

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:

$$browsing\_score(d|Q) = \log p(d) + \sum_{i=1}^{|Q|} \alpha^{i-1} \log p(w_i|d) \quad (16)$$

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

3.3 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:

$$p(w|Q) = \beta \log p(w) + \sum_{w' \in Q} \log p(w|w') \quad (17)$$

In our experiments we set the parameter β to 0.5.

4. 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 1: Dataset size

Table 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 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 is over 900 thousand which makes an average of 3 people using each tag.

Figure 3 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.

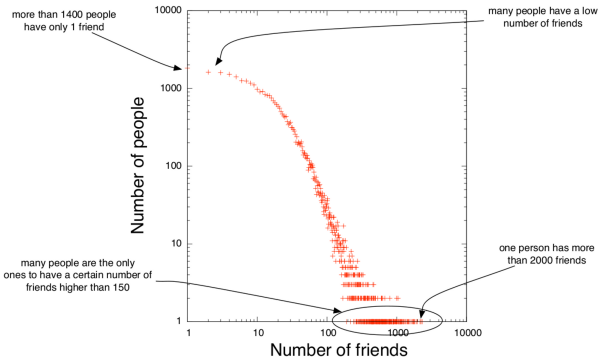


Figure 3: Plot of the distribution of friends.

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 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.

5. EVALUATION

We built a web application to evaluate our models in a user study. We 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.

5.1 Pilot Study

The pilot 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. Tracks are selected randomly from a pool of the 200 most popular tracks. The tag generation method is also selected randomly for each task.

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 are stored in a log in the database.

Most participants had fun during the experiment. Probably listening to the music and discovering new music helps with this fun aspect and keeps the participants motivated. A participant noticed that quickly he was selecting popular tags and quickly browsing for the "red link" to stop the task. This technique had him finish with the second place, we believe the first finishing participant had the same technique and was rejecting tasks 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 before hand 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 note "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". 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 2: 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. Participants completed more often tasks involving the social tag cloud rather than the two other tag clouds. Table 2 summarises the number of started and completed tasks and gives the relative frequency in percentage for each model. The relative frequency of completed tasks regarding the number of started tasks for each model is similar.

Figures 4 and 5 give an overview of the results. Figure 4 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. 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.

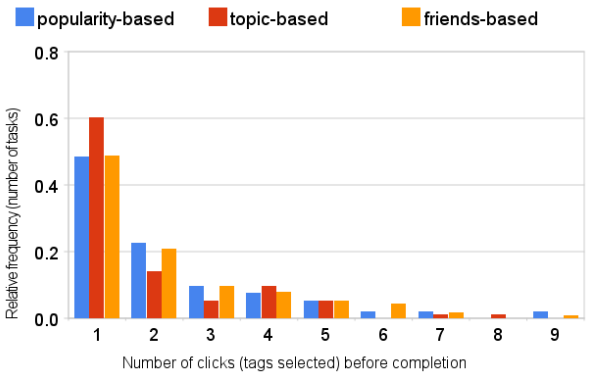


Figure 4: Histogram of different navigation path lengths across the three cloud generation models.

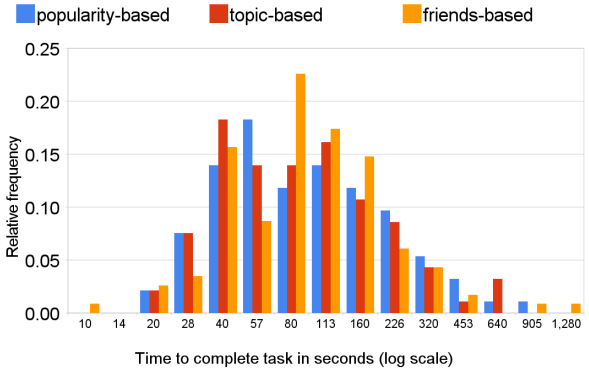


Figure 5: Histogram of time taking to complete tasks for different models.

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.

5.2 User survey

We conducted a short user survey together with the pilot study. Table 3 gives the statements that were asked to be ranked on a likert scale. Figure 6 represents the answers of the participants for each question.

The answers to question 1 clearly shows that our users are heavy internet users which you would expect when conducting a survey in a computer science faculty. Eleven partic-

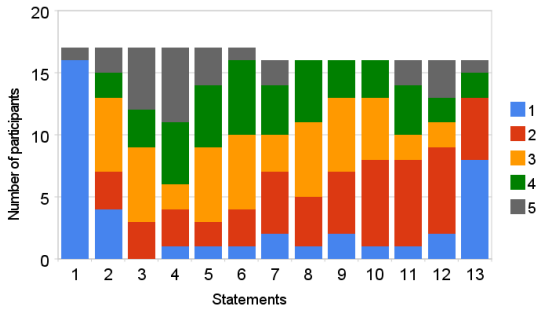


Figure 6: Number of participants per statement (best viewed in colour).

	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 3: Study statements

ipants 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.

5.3 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 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 4: 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 4 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 Experimenting with recommendation

The recommendation experiment consisted of tasks in which participants had to select a tag 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.

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: 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).

Table 5 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 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 6: 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, intu-

itively popular items are liked and already known, which is why they are popular because so many people know them. Table 6 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 personnalized, *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.

6. CONCLUSION AND FUTURE WORK

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.

6.1 Future Work

We are working on a new evaluation methodology to leverage the social model with social network data from the participants. The rest of the evaluation works as the one described in this paper. We believe that this personalized social model will outperform the topic model.

7. REFERENCES

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

[2] 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.

[3] P. R. Christopher D. Manning and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

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

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

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

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

[8] 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.

[9] 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.

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

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

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

A Method for Obtaining Semantic Facets of Music Tags

Mohamed Sordo
Universitat Pompeu Fabra
Barcelona, Spain
mohamed.sordo@upf.edu

Fabien Gouyon
INESC Porto
Porto, Portugal
fgouyon@inescporto.pt

Luís Sarmento
LIACC/FEUP, Univ. do Porto
Porto, Portugal
las@fe.up.pt

ABSTRACT

Music folksonomies have an inherent loose and open semantics, which hampers their use in structured browsing and recommendation. In this paper, we present a method for automatically obtaining a set of semantic facets underlying a folksonomy of music tags. The semantic facets are anchored upon the structure of the dynamic repository of universal knowledge Wikipedia. We illustrate the relevance of the obtained facets for the description of tags.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Dictionaries, Linguistic processing*; H.5.5 [**Information Storage and Retrieval**]: Sound and Music Computing

General Terms

Algorithms, Experimentation, Languages

Keywords

Music tagging, Last.fm, Wikipedia, Social music

1. INTRODUCTION

Music is a complex phenomenon that can be described according to multiple *facets*. Descriptive facets of music are commonly defined by experts (e.g. stakeholders in the music industry) in professional taxonomies. Multifaceted descriptions are especially useful for music browsing and recommendation. For instance, recommendations of the Pandora Internet radio use around 400 music attributes grouped in 20 facets,[1] as for instance Roots (e.g. "Afro-Latin Roots"), Instrumentation (e.g. "Mixed Acoustic and Electric Instrumentation"), Recording techniques (e.g. "Vinyl Ambience"), or Influences (e.g. "Brazilian Influences").

[1] `http://en.wikipedia.org/wiki/List_of_Music_Genome_Project_attributes`

WOMRAD 2010 Workshop on Music Recommendation and Discovery, colocated with ACM RecSys 2010 (Barcelona, SPAIN)

However, there exists no consensual taxonomy for music. Previous research showed the music industry uses *inconsistent* taxonomies [6], even when restricting to a single and widespread facet such as the music genre. Also, expert-defined taxonomies (music-related or not) have two fundamental problems. First, they are very likely to be *incomplete*, since it is impossible for a small group of experts to incorporate in a single structure all the knowledge that is relevant to a specific domain. Second, since domains are constantly evolving taxonomies tend to become quickly *outdated* –in music, new genres and techniques are constantly emerging.

An alternative strategy for describing music consists in relying on the broadness of the web and making use of the "wisdom of the crowds". Many music websites allow users themselves to assign their own descriptive tags to music items (artists, albums, songs, playlists, etc.). For instance, users of the website Last.fm tagged the band Radiohead as "90s", "00s", "alternative", "post-punk", "britpop", "best band ever", among other things. The combination of annotations provided by thousands of music users leads to the emergence of a large body of domain-specific knowledge, usually called *folksonomy*. Due to its informal syntax (i.e. direct assignment of tags), the tagging process allows the collective creation of very rich tag descriptions of individual music items.

When compared to taxonomies defined by experts, music folksonomies have several advantages. First, completeness, they ideally encompass all possible "ways to talk about music", including both *lay* and *expert* points of view. Second, due to the continuous nature of the tagging process, folksonomies tend to be well updated. Third, they usually incorporate both *commonly accepted* and *generic* concepts, as well as *very specific* and *local* ones.

It seems reasonable to assume that folksonomies tend to encompass various groups of tags that should reflect the underlying semantic facets of the domain including not only traditional dimensions (e.g. instrumentation), but also more subjective ones (e.g. mood). However, the simplicity and user-friendliness of community-based tagging imposes a toll: there is usually no way to *explicitly* relate tags with the corresponding music facets. For instance, a user may assign a number of tags related with music genre without ever actually explicitly specifying that they are about "music genre". For providing a flexible browsing experience, this is a significant disadvantage of folksonomy-based classification in relation to classification based on taxonomies, where the information about which facets are being browsed can be made

explicitly available to the user.

In this paper, we approach an essential research question that is relevant to bridging this gap: Is it possible to *automatically* infer the semantic facets inherent to a given music folksonomy? A related research question is whether it is then possible to classify elements of that music folksonomy with respect to the inferred semantic facets?

We propose an automatic method for (1) uncovering the set of semantic facets implicit to the tags of a given music folksonomy, and (2) classify tags with respect to these facets. We anchor semantic facets on metadata of the semi-structured repository of general knowledge Wikipedia. Our rationale is that as it is dynamically maintained by a large community, Wikipedia should contain *grounded* and *updated* information about relevant facets of music, in practice.

2. RELATED WORK

Music tags have recently been the object of increasing attention by the research community [3, 4]. A number of approaches have been proposed to associate tags to music items (e.g. a particular artist, or a music piece) based on an analysis of audio data [1, 9], on the knowledge about tag co-occurence [5], or on the extraction of tag information from community-edited resources [8]. However, in most cases, such approaches consider tags independently, i.e. not as elements in structured hierarchies of different music facets. When hierarchies of facets are considered, they are usually defined *a priori*, and greatly vary according to authors. For example, [4] groups tags in the following facets: genre, locale, mood, opinion, instrumentation, style, time period, recording label, organizational, and social signaling.

To our knowledge, however, few efforts have been dedicated to the particular task of *automatically* identifying the relevant facets of music tags. In their work on inferring models for genre and artist classification, Levy et al. apply dimensionality reduction techniques to a data set of tagged music tracks in order to obtain their corresponding compact representations in a low-dimensional space [5]. They base their approach on tag co-occurrence information. Some emerging dimensions can be associated to facets such as Era (e.g. the dimension [90s]). However, most of the dimensions thus inferred are, in fact, a combination of diverse music facets, such as for example the dimension [guitar; rock], which includes concepts of instrumentation and of genre.

Cano et al. use the WordNet ontology to automatically describe sound effects [2]. Albeit the very large amount of concepts in WordNet, they report that it accounts for relatively few concepts related to sound and music, and propose an extension specific to the domain of sound effects. On the one hand, they illustrate that browsing can indeed be greatly enhanced by providing multifaceted descriptions of items. On the other hand however, it is our belief that, because of their necessary stability, existing ontologies are not the most adapted tool to describe domains of knowledge with inherent open and dynamic semantics, such as music.

3. METHOD

Our method consists in using metadata from Wikipedia to infer the semantic facets of a given music folksonomy. This is performed in two steps. In the first step, we specialize the very large network of interlinked Wikipedia pages to the specific domain of the music folksonomy at hand. This is done by maximizing the overlap between Wikipedia pages and a list of frequent tags from the folksonomy. As the resulting network still represents a very large number of nodes, in a second step, we focus on the most relevant ones (node relevance being defined as an intrinsic property of the network). This step also includes additional refinements.

3.1 Obtaining a Music-Related Network

Wikipedia pages are usually interlinked, and we use the links between two particular types of pages (i.e. *articles* and *categories*) to construct a music-related network. Concretely, we use the DBpedia knowledge base (`http://dbpedia.org/`) that provides structured, machine-readable descriptions of the links between Wikipedia pages (DBpedia uses the SKOS vocabulary, in its 2005 version).[2] In particular, we make use of two properties that connect pages in DBpedia: (1) the property *subjectOf*, that connect articles to categories (e.g. the article "Samba" is a *subjectOf* of the category "Dance_music", and (2), the property *broaderOf*, that connect categories in a hierarchical manner (e.g. the category "Dance" is a *broaderOf* of the category "Dance_music", which is a *broaderOf* of the category "Ballroom_dance_music").

We start from the seed category "Music" and explore its neighbourhood from the top down, checking whether connected categories can be considered relevant to the music domain. A category is considered relevant if it satisfies any of the two following conditions:

- It is a tag from the folksonomy, such as for example "Rock and Roll". (This condition will be referred to as *isMusical*);
- At least one of its "descendants" is a tag from the folksonomy *and* the substring "music" is included in the title or the abstract of the corresponding Wikipedia article. (This condition is further referred to as *isTextMusical*.)

The "descendants" of a category are fetched from DBpedia using the two connecting properties previously described. These descendants can be either "successors" (i.e. all direct *subjectOf* and *broaderOf* of this category), or successors of successors, and so on. This iterative search is limited by a maximum depth, empirically fixed to a value of 4. Indeed, experiments with smaller values yielded a significant reduction of the tag coverage, while experiments with greater values did not increase significantly the coverage.

If any of the previous conditions is satisfied, the category, its successors and their edges are added to the network. Otherwise, the category and all incident edges are removed. The algorithm proceeds iteratively (following a Breadth-First search approach) until no more categories can be visited. A summarized version of the method for obtaining a music-related network is described in algorithm 1.

3.2 Finding Relevant Facets

Once the network of music-related categories is built, the next step is to find the nodes that are potentially more relevant to the network than others.

We invert the direction of the edges of the network in order to point back in the direction of the most generic category, i.e. "Music", and we compute the PageRank of the

[2] `http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102/`

Data: $C = \emptyset$, a list of categories (a queue, initially empty); $N = (V, E)$, a directed network with a set of nodes V and a set of edges E (initially empty);
Result: N, network with music nodes;
$C \leftarrow C \cup$ "$Music$";
while $C \neq \emptyset$ **do**
  $c \leftarrow first\ element\ of\ C$;
  $C \leftarrow C - c$;
  if *(c isMusical)* $\vee$ *((at least one descendant of c isMusical)* $\wedge$ *(c isTextMusical))* **then**
    $N \begin{cases} V \leftarrow V \cup c \cup successors(c) \\ E \leftarrow E \cup edges\ between\ c\ and\ successors(c) \end{cases}$
    $C \leftarrow C \cup successors(c)$
  else
    $N \begin{cases} V \leftarrow V - c \\ E \leftarrow E - all\ edges\ incident\ in\ c \end{cases}$
  end
end

Algorithm 1: Pseudo-code for the creation of a network of music-related categories from Wikipedia.

resulting network. PageRank [7] is a link analysis algorithm that measures the relative relevance of all nodes in a network. In PageRank, each node is able to issue a relevance vote on all nodes to which it points to (thus the need for re-orienting the edges). The weight of the vote depends on the relevance of the voting node (i.e. relevant nodes issue more authoritative votes). The process runs iteratively, and (under certain conditions) converges to a stable relative ranking, where nodes to which more edges from other relevant nodes converge (directly or indirectly) are considered more relevant. For initializing the PageRank algorithm, we set the initial weight of each node to 0.

In order to capture general yet complementary facets of music, we aim at reducing semantic overlap as much as possible by applying the following filters:

Stub Filter: We remove all categories with substring "_by_" and "_from_". We noticed that many categories in Wikipedia are actually combinations of two more general categories, as for instance "Musicians_by_genres", which is halfways between "Musicians" and "Music_genres" (see also figure 1). Further, we also remove categories that include "_music(al)_groups" (e.g. "Musical_groups-_from_California" that has hundreds of connected categories, hence a high PageRank). Most of these categories are used as *stubs*, even sometimes explicitly so we also excluded categories with the word "stub".

Over-Specialization Filter: We exclude all categories that include lexically a more relevant category. Many relevant categories are *specializations* of other more relevant ones, this occurs mostly with concepts related to anglophone music, which are described in great detail in Wikipedia (e.g. "American_Musicians" includes "Musicians" that has a higher PageRank).

Tag Filter: We remove all categories that are tags. Our objective is to uncover music facets that are implicit to the tags that make up a folksonomy. In general, tags are *elements* of such facets, not the facets themselves.

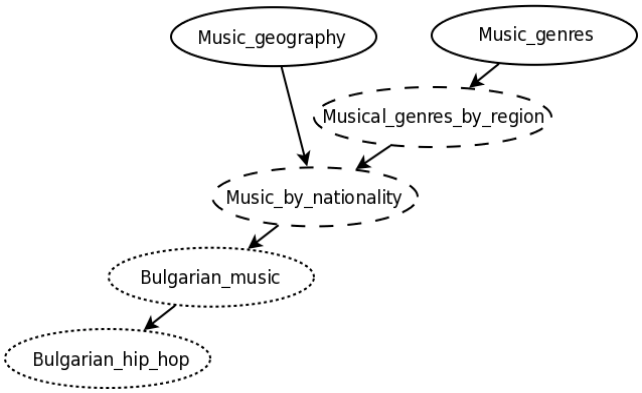


Figure 1: Example of subnetwork in our data. Dotted lines correspond to Wikipedia categories that are also Last.fm tags. Dashed lines correspond to categories not kept. Plain lines correspond to facets kept.

4. RESULTS

We experimented our method on a large dataset of artist tags, gathered from Last.fm during April 2010. The dataset consists of around 600,000 artists and 416,159 distinct tags. This dataset was cleaned in order to remove noisy/irrelevant data: (1) tags were edited in order to remove special characters such as spaces, etc.; (2) tags were filtered by weight[3], only tags with a weight ≥ 1 were kept; and (3) tags were filtered by popularity, keeping only tags with popularity ≥ 10, i.e. keeping only tags that were assigned to at least 10 artists. As a result, the final dataset consists of 582,502 artists, 39,953 distinct tags, and 9.03 tags per artist.

After running both stages of our method, we obtained a list of 333 candidate facets. Table 1 contains the top-50 facets, ordered by pagerank (top to bottom, left to right).

Table 1: Top-50 Wikipedia music facets

Music_genres	Aspects_of_music
Music_geography	Hip_hop_genres
Musical_groups	Music_of_California
Music_industry	Music_theory
Musicians	Rock_and_Roll_Hall_of_Fame_inductees
Musical_culture	Musical_subcultures
Occupations_in_music	Recorded_music
Music_people	Musical_quartets
Record_labels	Music_festivals
Music_technology	East_Asian_music
Sociological_genres_of_music	Centuries_in_music
Music_publishing_companies	Musical_composition
Musical_instruments	Musical_quintets
Anglophone_music	Southern_European_music
Music_of_United_States_subdivisions	Music_software
Western_European_music	Incidental_music
American_styles_of_music	Years_in_music
Radio_formats	Music_websites
Music_publishing	Guitars
Albums	Music_competitions
Musical_techniques	Musical_eras
Wiki_music	Music_and_video
Music_history	Musical_terminology
Music_performance	Music_halls_of_fame
Music_publishers_"people"	Dates_in_music

4.1 Assigning facets to tags

In order to assign a set of facets to a given Last.fm tag, we process the subnetwork of Wikipedia pages specialized to the Last.fm folksonomy (obtained in section 3.1), as described in algorithm 2 (Note that this process is restricted to tags that can be matched to one of the nodes in the network).

[3]i.e. Last.fm "relevance weight", which goes from 0 to 100

Table 2: Sample of the top tags for various music facets inferred

Music_genres	Occupations_in_music	Musical_instruments	Aspects_of_music
Sufi_music	Troubadour	Melodica	Rhythm
Dance_music	Bandleaders	Tambourine	Melody
Indietronica	Pianist	Drums	Harmony
Minimalism	Singer-songwriter	Synthesizers	Percussion
Singer-songwriter	Flautist	Piano	Chords
Music_software	**Music_websites**	**Music_competitions**	**Musical_eras**
Nanoloop	Mikseri.net	Nashville_Star	Baroque_music
Scorewriter	PureVolume	American_Idol	Ancient_music
MIDI	Allmusic	Melodifestivalen	Romantic_music
DrumCore	Jamendo	Star_Search	Medieval_music
Renoise	Netlabels	Eurovision_Song_Contest	Renaissance_music

Data: $C = \emptyset$, a list of categories (initially empty); F, a list of top-N music facets; t, a Last.fm tag;
Result: TF, list of facets applied to tag t;
$iter \leftarrow 1$;
$TF = \emptyset$;
while $(F \neq \emptyset) \vee (iter \leq maxIter)$ **do**
  $C \leftarrow C \cup predecessors(t)$;
  if $(\exists f \in (F \cap C))$ **then**
    $TF \leftarrow TF \cup f$
    $F \leftarrow F - f$
  end
  $iter \leftarrow iter + 1$
end

Algorithm 2: Pseudo-code for assigning Wikipedia facets to Last.fm tags

Given a Last.fm tag t, we look at its "predecessor" categories c, or more formally:

$$predecessors(t) = \{c|(t\ broaderOf(c)) \vee (t\ subjectOf(c))\}.$$

If any of these predecessors is a top-N facet, it is then assigned to t. The process continues iteratively until no more facets can be assigned to the tag, or a maximum number of iteration ($maxIter$) is exceeded. We empirically set this value to 8. This condition can be interpreted as the maximum distance in the network between a tag and a facet.

Table 2 presents a small subset of the obtained facets, followed by a subset of their corresponding list of top tags. Top tags are chosen based on the distance (in number of successive edges in the music network) to the given facet.

The relevance R_{tf} of a music facet f to a tag t is computed as the normalized inverse distance d_{tf} –in number of successive edges– between t and f:

$$R_{tf} = \frac{\frac{1}{d_{tf}}}{\sum_i \frac{1}{d_{ti}}}$$

For example, in figure 1, given the tag *bulgarian hip-hop*, our method starts navigating through the predecessors of this tag until finally reaching two music facets: *Music_genres* and *Music_geography*:

```
bulgarian hip-hop: {(Music_genres, 0.4),
                    (Music_geography, 0.6)}
```

5. SUMMARY AND FUTURE WORK

Although potentially more complete and up-to-date than taxonomies, music folksonomies lack structured categories, a particularly relevant aspect to browsing and recommendation. In this paper, we addressed the problem of uncovering the underlying semantic facets of the Last.fm folksonomy, using Wikipedia as backbone for semi-structured semantic categories.

There are many avenues for future work. First and foremost, we intend to evaluate the relevance of the obtained facets via systematic evaluations of tag classification. We will also study the distributions of music facets with respect to artist popularity. Further work should also relate to evaluating the usefulness of the obtained facets in a number of tasks, such as music recommendation, or tag expansion. We also intend to release the data (and code used to obtain it) in order to stimulate its use by fellow researchers.

6. ACKNOWLEDGMENTS

Thanks to Òscar Celma (BMAT), Eduarda Mendes Rodrigues (FEUP) and anonymous reviewers for useful comments. This work was partly supported by the *Ministerio de Educación* in Spain, and the *Fundação para a Ciência e a Tecnologia* (FCT) and QREN-AdI grant for the project Palco3.0/3121 in Portugal.

7. REFERENCES

[1] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere. Autotagger: A model for predicting social tags from acoustic features on large music databases. *JNMR*, 37(2):115–135, 2008.

[2] P. Cano, M. Koppenberger, P. Herrera, O. Celma, and V. Tarasov. Sound effect taxonomy management in production environments. In *AES*, 2004.

[3] O. Celma. *Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer, 2010.

[4] P. Lamere. Social tagging and Music Information Retrieval. *JNMR*, 37(2):101–114, 2008.

[5] M. Levy and M. Sandler. Learning latent semantic models for music from social tags. *JNMR*, 37(2):137–150, 2008.

[6] F. Pachet and D. Cazaly. A taxonomy of musical genres. In *RIAO*, 2000.

[7] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[8] L. Sarmento, F. Gouyon, and E. Oliveira. Music artist tag propagation with wikipedia abstracts. In *ECIR-WIRSN*, 2009.

[9] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE TASLP*, 2(16):467–476, 2008.

A Survey of Music Recommendation Aids

Pirkka Åman and Lassi A. Liikkanen
Helsinki Institute for Information Technology HIIT
Aalto University and University of Helsinki
Tel. +358 50 384 1514
firstname.lastname@hiit.fi

ABSTRACT

This paper provides a review of explanations, visualizations and interactive elements of user interfaces (UI) in music recommendation systems. We call these UI features "recommendation aids". Explanations are elements of the interface that inform the user why a certain recommendation was made. We highlight six possible goals for explanations, resulting in overall satisfaction towards the system. We found that the most of the existing music recommenders of popular systems provide no explanations, or very limited ones. Since explanations are not independent of other UI elements in recommendation process, we consider how the other elements can be used to achieve the same goals. To this end, we evaluated several existing music recommenders. We wanted to discover which of the six goals (transparency, scrutability, effectiveness, persuasiveness, efficiency and trust) the different UI elements promote in the existing music recommenders, and how they could be measured in order to create a simple framework for evaluating recommender UIs. By using this framework designers of recommendation systems could promote users' trust and overall satisfaction towards a recommender system thereby improving the user experience with the system.

Categories and Subject Descriptors

H5.m. Information interfaces and presentation: Miscellaneous. H.5.5 Sound and Music Computing.

Author Keywords

Recommendation systems, music recommendation, explanations, user experience, UI design.

1. INTRODUCTION

Recommender systems are a specific type of information filtering technique that aims at presenting items (music, news, other users, etc.) that user the might be interested in. To do this, information about the user is compared to reference characteristics, e.g. information on the other users of the system (collaborative filtering) or content features, such as genre in the case of books or music (content-based filtering). In its most common formulation, the recommendation task is reduced to the problem of estimating relevance of the items that a user has not encountered yet, and then presenting the items that have the highest estimated ratings [6]. The importance of recommender systems lies in their potential to help users to more effectively identify items of interest from a potentially overwhelming set of choices [7]. The importance of these mechanisms has become evident as commercial services over the Internet have extended their catalogue to dimensions unexplorable to a single user. However, the overwhelming numbers of content create a constant competition and can reduce the usefulness of recommendations unless they can persuade the user to try the suggested content. Explanations and other recommendation aiding UI features are examined in this paper as a way to increase the satisfaction towards recommenders among users.

The first interactive systems to have explanations were expert systems, including legal and medical databases [4]. Their present successors are commercial recommendation systems commonly found embedded in various entertainment systems such as iTunes [9] or Last.fm [12]. Explanations can be described as textual information telling e.g. why and how a recommendation was produced to the user. Earlier research shows that even rudimentary explanations build more trust towards the systems than the so-called "black box" recommenders [13]. Explanations also provide system developers a graceful way for handling errors that recommender algorithms sometimes produce [6].

The majority of previous recommendation system research has been focused on the statistical accuracy of the algorithms driving the systems, with little emphasis on interface issues and the user experiences [13]. However, it has been noted lately that when the new algorithms are compared to the older ones, both tuned to the optimum, they all produce nearly similar results. Researchers have speculated that we may have reached a level where human variability prevents the systems from getting much more accurate [7]. This mirrors the human factor: it has been shown that users provide inconsistent ratings when asked to rate the same item several times [14]. Thereby an algorithm cannot be more accurate than the variance in the user's ratings for the same item.

An important aspect for the assessment of recommendation systems is to evaluate them subjectively, e.g. how well they can communicate their reasoning to users. That's why user interface elements such as explanations, interactive elements and visualizations are increasingly important in improving user experience. In the past years subjectively perceived aspects of recommendations systems have accordingly gained ground in their evaluation.

In this paper we want to illustrate the possibilities of user-evaluation of recommendation supporting features in recommendation systems. We do this by performing a review on several publicly available music recommenders. Music is today one of the most ubiquitous commodities and the availability of digital music is constantly growing. Massive online music libraries with millions of tracks are easily available in the Internet. However, finding new and relevant music from those vast collections of music becomes similarly increasingly difficult. One approach to tackle the problem of finding new, relevant music is developing better (reliable and trustworthy) recommendation systems. Music recommenders are also easy to access and music has reasonably short process in determining the quality of recommendation results.

WOMRAD 2010 Workshop on Music Recommendation and Discovery, colocated with ACM RecSys 2010 (Barcelona, SPAIN)

2. GOALS FOR RECOMMENDATION AIDS

Tintarev and Masthoff [16] present a taxonomy for evaluating goals for explanations. Those are shown slightly modified in the Table 1 below. We argue that satisfaction towards a recommendation system is an aggregate of the six other dimensions, more a goal of itself than the other dimensions. In addition, we noticed that the dimensions are not so straightforward as Tintarev and Masthoff present them. Some of them cannot be evaluated using objective measures, and therefore framework for evaluation recommendation aids must be drawn from user research. In the following we describe each dimension and give examples of how they could be evaluated and measured.

Table 1. Dimensions for recommendation explanations.

Goal	Definition
Transparency	Explain how the system works
Scrutability	Allow users to tell the system it is wrong
Effectiveness	Help users make good decisions
Persuasiveness	Convince users to try or buy
Efficiency	Help users make decisions faster
Trust	Increase users' confidence in the system

Resulting in →

Satisfaction (increasing the ease of use or enjoyment towards the system)

1. An explanation may tell users how or why a recommendation was made, allowing them to see behind the UI and thus making recommendation *transparent*. Transparency is also a standard usability principle, formulated as a heuristic of 'Visibility of System Status' [13]. Transparency can be measured objectively, using binary scale (yes/no), e.g. if a UI provides some kind of explanation how a recommendation was made transparency gets a vote. However, evaluating transparency subjectively may involve users to be asked if they understand how the recommendation was made using e.g. Likert scale.

2. *Scrutability* means that users are allowed to provide feedback for the system about the recommendations. Scrutability is related to the established usability principle of 'User Control' [13]. Scrutability can be measured objectively by finding out if there is a way to tell the system it is wrong. To evaluate scrutability subjectively, users may be given a task to find a way to tell how to stop receiving e.g. recommendations of Elvis songs. If users feel they can control the recommendations by changing their profile, the UI has the possibility to scrutinize.

3. *Effectiveness* of an explanation help users make better decisions. Effectiveness is highly dependent on the accuracy of the recommendation algorithm. An effective explanation would help the user evaluate the quality of suggested items according to their own preferences [16]. This would increase the likelihood that the user discards irrelevant options while helping them to recognize useful ones. Unlike travel or film recommenders, in the case of music recommenders the process of deciding the goodness of a recommendation is done quite quickly.

4. *Persuasiveness*. Explanations may convince users to try or buy recommended items. However, persuasion may result in an adverse reaction towards the system, if users continuously decide to choose bad recommendations. Persuasion could be measured according to how much the user actually tries or buys items compared to the same user in a system without an explanation facility [16] and what kind of persuasion techniques are utilized. Persuasion could also be measured by applying click-through rates used in measuring online ads.

5. *Efficient* explanations help users to decide faster which recommendation items are best for their current situation. Efficiency can be improved by allowing the user to understand the relation between recommended options [12]. A simple way to evaluate efficiency is to give users tasks and measure how long it takes to find e.g. an artist that is novel and pleasing to the user.

6. Increasing users' confidence in the system results in *trust* towards a recommender. Trust is in the core of any kind of recommendation process, and it is perhaps the most important single factor leading to better user satisfaction and user experience with the interactive system. A study of users' trust (defined as perceived confidence in a recommender system's competence) suggests that users intend to return to recommender systems, which they find trustworthy [2]. The interface design of a recommender affects its credibility and earlier research has shown that in user evaluation of web page credibility the largest proportion of users' comments referred to the UI design issues [5]. Trust needs to be measured using subjective scales over multiple tasks or questions about the recommendation aiding features of a recommender UI.

The ease of use or enjoyment results finally in more *satisfaction* towards a system. Descriptions of recommended items have been found to be positively correlated with both the perceived usefulness and ease of use of the recommender system [6], enhancing users' overall satisfaction. Even though we see satisfaction as an aggregate of the dimensions presented above, satisfaction with the process could be measured e.g. by conducting a user walk-through for a task such as finding a satisfactory item.

3. RELATED EMPIRICAL RESEARCH

It is widely agreed that expert systems that act as decision-support systems need to provide explanations and justifications for their advice [13]. However, there is no clear consensus on how explanations should be designed in conjunction with other UI elements or evaluated by users. Studies with search engines show the importance of explanations. Koenmann & Belkin [11] found that greater interactivity for feedback on recommendations helped search performance and satisfaction with the system. Johnson & Johnson [10] note that explanations play a crucial role in the interaction between users and interactive systems. According to their research, one purpose of explanations is to illustrate the relationship between cause and effect. In the context of recommender systems, understanding the relationship between the input to the system (ratings and choices made by user) and output (recommendations) allows the user to interact efficiently with the system. Sinha and Swearingen [15] studied the role of transparency in recommender systems. Their results show that users like and feel more confident about recommendations that they perceive transparent. Explanations allow users to meaningfully revise the input in order to improve recommendations, rather than making "shots in the dark."

Herlocker and Konstan [6] suggest that recommender systems have not been used in high-risk decision-making because of a lack of transparency. While users might take a chance on an opaque movie recommendation, they might be unwilling e.g. to commit to a vacation spot without understanding the reasoning

behind such a recommendation. Building an explanation facility into a recommender system can benefit the user in various ways. It removes the "black box" around the recommender system, providing transparency. Some of the other benefits include *justification*. If users understand the reasoning behind a recommendation, they may decide how much confidence to place in the suggestion. That results in greater *acceptance* or *satisfaction* of the recommender system as a decision aide, since its limits and strengths are more visible and its suggestions are justified.

4. RECOMMENDATION AIDS IN EXISTING MUSIC RECOMMENDERS

We conducted an expert walkthrough of six publicly available music systems with recommendation functionalities in order to find out which of the six goals explanations, visualizations and interactive UI elements promote in the existing music recommenders, and how they can be measured in order to create a simple framework for evaluating recommenders. The walkthrough was conducted by authors listing the UI features capable of promoting the goals mentioned above. The reviewed systems include Pandora, Amazon, Last.fm, Audiobaba, Musicovery and Spotify. We wanted to include the most popular online music services, and on the other hand, include a variety of different UIs. Each of the evaluated systems provides recommendations but not necessarily explanations. Systems without textual explanations were also included in order to find out what kind of goals or functions similar to verbal explanations other recommendation aids provide.

Table 2. The occurrences of recommendation aids in a selection of music recommenders

	Trans.	Scrt.	Effect.	Pers.	Effic.	Trust	
Amazon	1	2	2	3	1	3	12
Last.fm	-	2	2	1	2	2	9
Audiobaba	1	1	2	1	1	2	8
Musicovery	2	2	2	2	2	1	11
Spotify	-	-	1	1	1	1	4
Pandora	2	2	3	3	2	3	15
	6	9	12	11	9	12	

If a recommender has a possibility to promote a goal with explanations, visualizations or interactive elements, it gets a vote in Table 2. For example, persuasiveness promoted through visualizations is potentially possible in all of the interfaces that have visualizations, even rudimentary ones, such as an album cover. A single user might be persuaded to try or buy by presenting a subjectively compelling album cover. From Table 2 we can see that Pandora, Amazon and Musicovery have the greatest number of UI elements able to provide users support for sense-making of recommendations. Effectiveness, persuasiveness and trust are the most commonly promoted goals. In each recommender, each UI element has the potential to increase trust towards the systems, but for more accurate measurement, it remains to be evaluated by empirical user research, to which extent each elements in certain recommender interface really promote trust. This applies to most of the six goals: without empirical data, it is almost impossible to decide, whether the potential for promoting effectiveness, persuasiveness and efficiency actually realizes. Only transparency and scrutability can be measured using objective binary scale of yes/no, but they can be evaluated also using subjective (Likert style) scales. We argue that by measuring these goals for UI elements together with a set of usability guidelines, it is possible to evaluate and design better user experiences for recommendation systems.

Some of the dimensions are easy to connect to certain UI elements. For instance, scrutability is usually designed as a combination of explanation and interactivity, whereas other, more general level dimensions depend strongly on subjective experience and are hard to connect with specific UI elements. For example, satisfaction or trust towards a system is usually combination of different experienced UI dimensions. Therefore the most common dimensions promoted in the evaluated systems were trust and satisfaction. Those, together with persuasiveness, are experienced very subjectively, which means that empirical user evaluation is needed for more reliable and comparable evaluations of those dimensions.

Obvious example of an explanation providing transparency is Amazon's "Customers with Similar Searches Purchased…", with up to ten albums' list. Pandora tells a user: "This song was recommended to you because it has jazzy vocals, light rhythm and a horn section." Transparency is very hard to achieve without textual, explicit explanations. Of the reviewed systems, only Musicovery's UI with several interactive elements, graphical visualization of the recommendations and the relations between them give users clear clues of why certain pieces of music were recommended, without providing explanations.

Last.fm offers users scrutability in many ways, e.g. with its music player (Figure 1). One of the system's more sophisticated scrutinizing tactics is a social one. Last.fm allows users to turn off the registering (called scrobbling) of the listened music. The system's users can perform identity work by turning scrobbling off, if they feel they do not want to communicate what they have listened to the other users. Amazon provides "Fix this recommendation" option for telling the system to remove recommended item from the users browsing history.

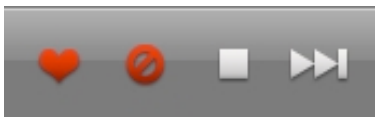


Figure 1: Example of scrutable interactivity: Last.fm player's love, ban, stop and skip buttons give users a tool to control their profiles and thereby affect recommendations.

Users can be helped in efficiency and effectiveness, i.e. making better and faster decisions by offering appropriate controls with interactive elements. For instance, Musicovery's timeline slider is presented in Figure 2. It works in real time with the system's graphical presentation of recommended items.

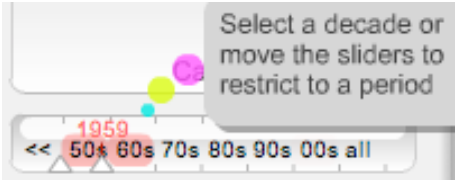


Figure 2: Musicovery's timeline slider: interactivity promoting efficiency, scrutability, and effectiveness, resulting in more trust and satisfaction towards the system.

5. DISCUSSION AND CONCLUSIONS

We reviewed dimensions of explanations in six music recommendation systems and found out that most of the reviewed commercial music recommendation systems are "black boxes", producing recommendations without any, or

very limited explanations. Most of the dimensions are poorly promoted by textual explanations, but can be promoted by other means, namely by visualizations and interactive elements, and further, by user-generated content and social facilities. From the expert walkthrough of the selected music recommendation systems we can draw a tentative conclusion that if UI elements can fulfill similar functions as explanations, there is necessarily no need for textual descriptions. By using non-verbal recommendation aids as "implicit" explanations and using them in recommendation system design, we can promote better user experience. This is the case especially when the user has enough cultural capital and therefore competence for "joining the dots" between recommended items without explicit explanations. On the other hand, if the recommender is used e.g. for learning about musical genre, textual explanations may be indispensable.

As an example of the dimensions that UI elements other than verbal explanations can promote is the overall satisfaction or trust towards the systems that can be achieved by conversational interaction such as in UI example presented in Figure 3, where users are given a chance for optional recommendations based on their situational desires and needs.

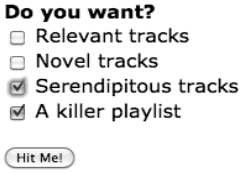


Figure 3: A recommendation aid with optional inputs.

Last.fm is an example of recommendation system with no explanations. However, it has an abundance of other elements such as user created biographies, genre tags and pictures of artists, not to mention advanced social media features that together effectively work towards the same goals as the dimensions of explanations. Furthermore, Spotify, a popular European music service with very simple recommendation facility, does not provide any explanations whatsoever. Its popularity relies on providing users a minimalistic UI with effective search facility and a functional, high-quality audio streaming. Spotify's usability and functionality work effectively towards overall satisfaction of the system, making explanations, visualizations or advanced interactivity redundant. Obviously, Spotify's abilities for helping to find new music are limited, because of very simple recommendation facility, but it can be used as an example of the argument that user trust and satisfaction can be promoted by diverse means depending on the different users' various needs and desires.

The next step of our research is to conduct an empirical user evaluation of the importance and functions of different UI elements in music recommenders. We are looking for feasible scales of measurement that are drawn from user evaluation of the goals for UI elements in recommenders. User evaluation could be done with modified music recommender UIs where users are given tasks and comparing e.g. how much taking away a UI feature such as an explanation effects to the time the task is completed. It would also be interesting to explore how different goals can be promoted by combining various UI elements, and by assigning unconventional roles for UI elements, e.g. creating visualizations that would reveal the logic behind a recommendation and at the same time give a user a tool to scrutinize.

REFERENCES

[1] Adomavicius G., Tuzhilin, A. 2005. Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749.

[2] Buchanan, B., Shortcliffe, E. 1984. *Rule-Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*. Reading, MA: Addison Wesley Publishing Company.

[3] Chen, L., P. Pu. 2002. Trust building in recommender agents. In Proc. of *International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces '02*.

[4] Doyle, D., A. Tsymbal, and P. Cunningham. 2003. *A review of explanation and explanation in case-based reasoning*. Technical report, Dept. of Computer Science, Trinity College, Dublin.

[5] Fogg, B. J., Soohoo, C., Danielson, D. R., Marable, L., Stanford, J, Tauber, E.R. 2003. How do users evaluate the credibility of web sites? In Proc. of *Designing for User Experiences '03*. Pages 1-15.

[6] Herlocker J. L., Konstan, J. A. 2000. Explaining collaborative filtering recommendations. Proc. of *Computer Supported Collaborative Work '00*. Pages 241-250.

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

[8] Hill, W., Stead, L., Rosenstein, M., Furnas, G. 1995. "Recommending and Evaluating Choices in a Virtual Community of Use", Proc. of *Conference on Human Factors in Computing Systems '05*.

[9] iTunes, http://www.apple.com/itunes.

[10] Johnson, J. & Johnson, P. 1993. Explanation facilities and interactive systems. In Proceedings of *Intelligent User Interfaces '93*. (159-166).

[11] Koenemann, J., Belkin, N. 1996. A case for interaction: A study of interactive information retrieval behavior and effectiveness. In Proc. *of Conference on Human Factors in Computing Systems '96*, ACM Press, NY.

[12] Last.fm, http://www.last.fm.

[13] Nielsen, J. and R. Molich. Heuristic evaluation of user interfaces. In Proc. of *Conference on Human Factors in Computing Systems '90.*

[14] Pu, P., Chen, L. 2006. Trust building with explanation interfaces. In Proc. of *Intelligent User Interfaces '06*, pages 93-100.

[15] Sinha, R., Swearingen, K. 2002. The role of transparency in recommender systems. In Proc. of *Conference on Human Factors in Computing Systems '02*.

[16] Tintarev, N., Masthoff, J. 2007. Survey of explanations in recommender systems. In Proc. of *International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces '07*.

The Role People Play in Adolescents' Music Information Acquisition

Audrey Laplante
École de bibliothéconomie et des sciences de sciences de l'information, Université de Montréal
C.P. 6128, succ. Centre-ville, Montréal QC, Canada
audrey.laplante@umontreal.ca

ABSTRACT
This paper reports on a study on the role people play in music information provision for adolescents. Using a qualitative approach to social network analysis, this study focuses on the ways in which music information is shared across adolescents' networks. Preliminary findings suggest that adolescents primarily discover new music through close friends whose social network is significantly different from theirs (e.g., those who attend a different school). They also indicate that music opinion leaders (i.e., those who are most influential in their social network in terms of music) are perceived as (1) good communicators, who are (2) highly invested in music, and who are (3) willing to share the information with their friends. These findings provide developers with ideas for the improvement of social filtering algorithms used in music recommender systems.

Categories and Subject Descriptors
H.1.2. [**Models and Principles**]: User/Machine Systems---human factors; H.5.5. [**Information Interfaces and Presentation**]: Sound and Music Computing---Systems.

General Terms
Human Factors.

Keywords
Social networks, music information behavior, adolescents, music recommender systems, user studies.

1. INTRODUCTION
For many years, it has been common practice for people, especially adolescents, to share music. While yesterday's young adults exchanged CDs and tapes, or prepared music compilations for one another, today's adolescents share music files through peer-to-peer file sharing systems and push music information to their friends using instant messaging or social networking sites. If the media have changed, the motivations remain unchanged: music sharing strengthens social bonds [1] and represents one of the most efficient ways of discovering new music [2, 3]. Indeed, acting as filters between music and their friends, people provide highly personalized recommendations, specially tailored to their friends' tastes.

Considering the effectiveness of people as sources of music recommendations, it comes with no surprise that when developers tried to automate the process of recommending music, many decided to use social filtering. As a matter of fact, although a few successful music recommender systems use content-based filtering (e.g. *Pandora*), most systems exploit feedback from other users to offer personalized recommendations (e.g. *Last.fm*). What makes each of these systems unique is the type of information they use to generate recommendations: implicit feedback (e.g., listening habits, purchases) and/or explicit feedback (e.g., user ratings, lists of favorite artists) [4].

By providing a rich description of the role people play in music information acquisition in adolescents' daily lives, this study contributes to our understanding of the music information behavior of young adults and highlights potential avenues for the development of more efficient collaborative filtering algorithms for music recommender systems.

WOMRAD 2010 Workshop on Music Recommendation and Discovery, colocated with ACM RecSys 2010 (Barcelona, SPAIN)

2. RELATED WORK

2.1 People as Information Providers
Research performed by information scientists has shown that people (relatives, friends, colleagues, and other acquaintances) play a primary role in information provision. This phenomenon seems to be particularly prevalent in everyday life contexts [6-8], for instance to acquire hobby-, health- or job-related information. People rely on their social network for information or recommendations for a variety of reasons. One's close personal network is usually considered the most accessible source of information. Family and close friends are generally close by and willing to share information, both spontaneously and on demand [9]. Additionally, people appreciate the relevance of the information they acquire this way. By asking people who know them well and whom they trust and consider to have good judgment, they obtain information that has been filtered especially for them [10]. Information sharing between individuals is also socially and emotionally motivated: (1) it helps build and maintain relationships, and (2) sharing information is a gratifying activity [11].

The few studies that have been conducted on music information behavior in everyday life revealed that people play perhaps an even more important role than in other contexts. Studies by Laplante [2] and Tepper & Hargittai [12] showed that one's social network represent the most important source of music discovery. Similarly, Sinha & Swearingen [13], who compared music recommendations provided by friends and online systems, found that the former consistently performed better than the latter from the user's perspective.

2.2 Music and Identity
If people rely so extensively on their social network to discover new music, it might be because of the close link that exists between music and identity. Research in music sociology and psychology has long established that music played a significant role in the formation of one's identity, particularly in adolescence [14, 15]. Young people use music as a social badge which communicates who they are (or who they wish they were) individually and as a group [16]. Indeed, adolescents' music tastes develop in a highly social environment: their music preferences are usually similar to

those of their friends or of people they wish to emulate. Thus, it is common practice for adolescents to scan the music collections of their most estimated friends to check for new suggestions, as well as to look at the collections of newcomers or potential love interests to ensure that they "fit" [15]. This also explains why sub-cultures, by which groups of adolescents often define themselves and to which opinions, attitudes and values are associated, usually form around music genres [16].

As a matter of fact, adolescents do not only use music tastes to express who they are but also to judge their peers. Hence, most consider that there are social repercussions associated with the fact of exposing their music preferences [15]. For instance, research revealed that those who express a preference for music genres that are considered prestigious by their peers are more likely to be perceived positively [14]. In the same way, those who demonstrate a high level of knowledge of popular music were found to have more chances of being perceived as popular by their peers [17]: music being one of the most important conversation topics among adolescents, one can assume that good music knowledge facilitates social interactions with others.

Considering that adolescents use music preferences to make inferences about others, it comes with no surprise that most seek to "perform" through their music tastes: the values and attitudes associated with the music genres they publicly admit liking must correspond to what they want to convey about themselves [14]. And the advent of social networking sites such as *MySpace* or *Facebook*, which allow them to list their interests in terms of music, cinema, television series and books on the social network profile, has emphasized this phenomenon [18].

3. METHODOLOGY

Qualitative methodologies have dominated the research on everyday life information behavior, with interviews, diaries and observation being the most common data collection methods. These methods have proven to be effective in providing thick descriptions of the phenomenon from the user's perspective and in capturing the richness of the context into which it occurs. For this project, a qualitative approach to social network analysis has been adopted. Social network analysis (SNA) focuses on "relationships among social entities, and on the patterns and implications of these relationships," [19] in particular on the flow of resources (e.g., information) among actors. It provides a set of techniques and theoretical concepts and properties researchers can use to analyze and describe social networks. First developed and employed by sociologists, it is now used in many other disciplines, including information science [20].

For this study, an egocentric approach to social networks has been adopted. This approach consists in examining the social network of focal persons (called "egos"). Egos are asked to name the persons which whom they maintain relationships (called "alters"). Egos are then asked to provide information about their ties to alters as well as about ties between alters in their social network [19]. Researchers also commonly ask proxy reports about alters [21].

3.1 Participants

This study is designed to run from May 2010 to April 2011 with the expectation of recruiting 25 participants. The population studied is composed of French-speaking adolescents (15-18 year-old) living in the Quebec province (Canada). This paper presents the preliminary findings derived from the six interviews conducted to date, which represent a total of 486 minutes of recording. Participants were selected following the maximum variation sampling strategy as described in [22]. Among the six participants, four were female. At the time of the interview, five were full-time high-school students and one was a full-time college student.

3.2 Data Collection and Analysis

Data are collected through in-depth interviews. Social network theory, together with a review of related works on everyday life information behavior and on music and identity, provided a useful theoretical background for the development of the data collection instrument. The resulting instrument is composed of an adaptation of the social network-mapping tool developed by Todd and described in [23], followed by a traditional interview schedule. The social network map is filled by the participant with the help of the researcher. To elicit the names to be included on the map, participants are asked to think about how they could group people around them (e.g., school, relatives, neighbors), to note these groups on the map, and then to name the persons they feel close to in each group (the alters). Participants are invited to place alters on the map, using the concentric circles to indicate the strength of their relationship with each of them, as well as to draw lines to indicate relationships between alters (the strength of the tie being represented by the thickness of the line). Participants are then asked to add on the map any other person with whom they share music information. They are asked (1) to draw a circle around those with whom they discuss music most often, (2) to mark with an asterisk the persons whom they trust the most for music recommendations, and (3) to draw a box around the name of those with whom they maintain a relationship essentially based on music (see figure 1 for an example of a social network map filled by a participant). Participants are requested to provide information about each alter and their relationship with them, including information about their music tastes, the nature of the music information they exchange with them if any, and the influence they have on their own music preferences and on those of their group. The interviews also include general questions on participants' music tastes and listening habits.

All interviews are recorded. Both the interview and the resulting social network map are transcribed into computer files. Interviews are analyzed using NVivo by QSR, a software package designed specifically for the encoding and analysis of qualitative data.

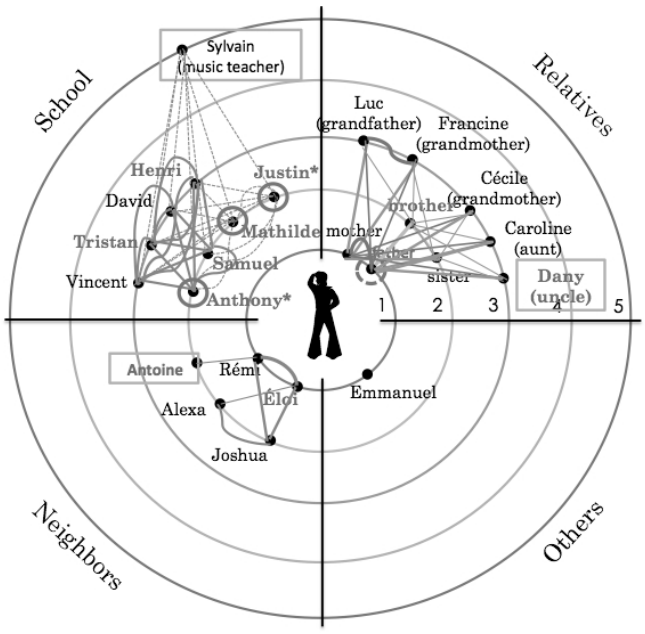


Figure 1. Example of a participant's social network map

4. FINDINGS

4.1 The Influence of Others on Music Tastes

4.1.1 Recent Changes in Music Tastes

In the introduction of the interviews, participants were asked if their music tastes had changed significantly in the last three years. Perhaps unsurprisingly considering the fact that adolescence is a period characterized by changes, all affirmed that their music tastes had greatly evolved. But the most interesting aspect of their answers resided in the reasons they gave to explain it: all were related to changes in their social network. One mentioned that she had discovered a new music genre because of her new boyfriend; three explained that they had changed school and, as a result, had made new friends through which they had discovered new music ("You start high school and the people, what they make you listen to, it's not the same type of music. And then, by listening to this music, you start liking it too."); and two were just not very interested in music before, but because it was such an important topic of discussion at school, music had slowly taken a more important place in their lives.

4.1.2 Opinion Leaders

"Opinion leaders" are defined as individuals who have developed an expertise in a specific domain. Because of this expertise, people are more likely to turn to them for information or recommendations, which makes them influential in their environment. During the interviews, participants were asked to identify who in their social network exerted more influence on others in terms of music. Three self-identified (two girls and one boy) as being an opinion leader for music in their group. Music opinion leaders were generally not considered as being influential in other domains. Indeed, in every densely knit group, each member seems to have his/her domain of influence, whether it is music, fashion, movies, television series or books. The analysis revealed that music opinion leaders were perceived as (1) good communicators, who are (2) highly invested in music, and who are (3) willing to share the information with their friends. Their desire to have unique knowledge and to be a resource person for others leads them to constantly look for new music. Hence, describing an influential friend, one participant says: "[Anthony] just got a satellite radio, so he picks up loads of stations. When they are playing a tune, he sends you the title. And then, he takes notes of everything. And when he finds a good song, he gives it to me." However, opinion leadership is only possible if one is surrounded by people who have similar music tastes. One participant, whose social network was mostly composed of heterophilous relationships, particularly in terms of music preferences, explained that her tastes were too unusual for an adolescent and therefore very unlikely to meet her friends' tastes. As a result, she did not feel it was relevant to share any music information with them.

4.2 The Strength of Weak Ties?

In a highly cited journal article, Granovetter proposed in 1973 the Strength of Weak Ties [24], a theory that has proven to be particularly useful to understand the role people play in information provision. According to this theory, weak ties (acquaintances) would be more instrumental than strong ties (friends and family) to obtain new information because of the high degree of overlap that generally exists between the social networks of strong ties. In other words, my close friends, who generally know the same people I know, are more likely to have access to information to which I also have access; whereas acquaintances, who usually have a social network significantly different from mine, are more likely to have access to different information [25]. Following this theory, we could expect that weak ties would be more useful to discover new music. However, participants' accounts suggest that weak ties might not be as instrumental in music discovery as they are in other contexts, such as when people are looking for a job.

4.2.1 The Role of Weak Ties

In some occasions, participants' accounts fit the theory perfectly. Indeed, two participants identified a few weak ties as being important sources of music discovery for them. In these cases, the weak ties were people who were much older than they were (their parents' age) and their relationship with them was mostly based on music (i.e., music is their main conversation topic when they meet). They were considered by participants as experts in music, sometimes for a specific music genre: one had extensive knowledge in classical music, two had very large music collections, another was an amateur musician who loved blues, and one was a music teacher specialized in jazz (the last two are represented in Figure 1 as Dany and Sylvain). Music information mostly flowed in one direction: while they admit being influenced by these people, they did not consider they were influential for them.

Weak ties, however, were not considered instrumental by the four other participants for the acquisition of music information. A possible explanation would be that music preferences are considered too personal and subjective to trust recommendations from someone one does not know well. What is more, considering that adolescents are conscious of being judged on their music preferences, following advices from weak ties might be considered too risky from a social point of view. But Granovetter's theory nevertheless provided an interesting theoretical framework to understand who in their network were more likely to represent a good source for discovering new music. According to the theory of the Strength of Weak Ties, weak ties are crucial in information provision because the overlap between their social network and the Ego's social network is less important than it is between strong ties. Although the main sources of music discovery were not weak ties, the strong ties from whom they were more likely to seek recommendations were almost always those whose social network were more different from theirs, mostly those who were going to a different school. For instance, one participant explained making new discoveries mainly through her friends with whom she skies "because we don't hang out with the same gang at school."

Exchanging music information with weaker ties also seemed to be socially motivated. Indeed, music seems to be to adolescents what weather is to adults: the default conversation topic. Hence, one participant reported that "at school, everybody has a iPod. So you're there, during lunchtime, and everybody has earplugs, so it's easy to stop and say 'hey, what tunes do you have?'" Music can also be at the origin of a relationship. For instance, one participant reports having realized through *Facebook* that some people she did not know well had music tastes that were similar to hers. This realization had led her to engage in conversations about music with them and, as a result, to become closer to them.

4.2.2 The Role of Strong Ties

A lesser-known aspect of Granovetter's theory concerns the value of strong ties in information provision. According to the theory, "strong ties have greater motivation to be of assistance and are typically more easily available" [25]. They also have greater influence and more credibility. This corresponds to the accounts provided by our participants. Strong ties, including those with whom they share most of their social network, play a crucial but different role in music information acquisition than weaker ties do. In addition to the role of strong ties in music discovery, it emerges from the analysis that music information was shared with strong ties also for two other reasons: (1) to maintain or reinforce a relationship, and (2) to legitimate our tastes. For instance, one

participant describes that she shares music (or lyrics) with her friends to show that she understands how they feel. Thus, depending on what they confide in her, she suggests music she believes will help them go through what they are experiencing. Another participant reports that when he discovers something new, he asks his best friend to listen to it so they can talk about it, which, he will weakly admit, helps him form an opinion about the music.

And which are the strong ties through which music information is exchanged? Of course, close friends were often mentioned. These close friends were usually those who have music tastes that are similar to theirs and who are perceived as having good judgment in terms of music (i.e., those who have good music knowledge and are considered to be "independent thinkers"). Hence, one participant reported trusting a friend who is a musician and explained: "There are people, you know they are into music, so it's really something they know well. So when they talk, you know they don't give you titles just to give you titles." The same participant later added that she did not trust one of her best friends, although they had similar music tastes: "[She] doesn't want to stand out. She does what everybody else does. She won't influence you, you're always the one who influences her. You tell her you like something, the day after, she will have downloaded it." If close friends were pivotal in the acquisition of music information, older siblings and even parents were too. As a matter of fact, all participants reported being influenced by at least one of their parents in terms of music. One participant explained: "The songs you grew up with, whether you want it or not, you always end up listening to them again." Another said: "I've heard [their music] so many times that I listen to it and I like it really." This supports the findings of music sociologists who found that familiarity often leads to appreciation [26].

5. CONCLUSION

Considering the small size of the sample, the results presented here should be interpreted carefully. However, by providing rich descriptions of the ways in which music is shared within the social networks of some adolescents, this study provides a first glance at the role people play in music information acquisition in adolescence, while shedding some light on the process through which young adults discover new music through friends, relatives or other acquaintances. Results suggest that further research on the characteristics of the structure and the ties of which social networks are composed, and the impact of these characteristics on the flow of music information could help inform the design of music recommender systems.

6. ACKNOWLEDGMENTS

This work has been supported by the Université de Montréal, the Social Sciences and Humanities Research Council of Canada and the Fonds québécois de la recherche sur la société et la culture.

7. REFERENCES

[1] T. DeNora, *Music in everyday life*. Cambridge; New York: Cambridge University Press, 2000.

[2] A. Laplante, "Everyday life music information-seeking behaviour of young adults: an exploratory study," Ph.D. thesis, McGill University, Montreal, QC, 2008.

[3] B. Brown, et al., "Music sharing as a computer supported collaborative application," 2001, pp. 179-198.

[4] O. Celma, "Foafing the music: Bridging the semantic gap in music recommendation," *The Semantic Web-ISWC 2006*, pp. 927-934, 2006.

[5] K. Swearingen and R. Sinha, "Beyond algorithms: An HCI perspective on recommender systems," 2001.

[6] D. E. Agosto and S. Hughes-Hassell, "People, places, and questions: an investigation of the everyday life information-seeking behaviors of urban young adults," *Library and Information Science Research*, vol. 27, pp. 141-163, 2005.

[7] H. Julien and D. Michels, "Intra-individual information behaviour in daily life," Information Processing & Management, vol. 40, pp. 547-562, 2004.

[8] J. Hartel, "The serious leisure frontier in library and information science: Hobby domains," *Knowledge organization*, vol. 30, pp. 228-238, 2003.

[9] K. Williamson, "Discovered by chance: the role of incidental information acquisition in an ecological model of information use," *Library and Information Science Research*, vol. 20, pp. 23-40, 1998.

[10] Y. Lu, "The human in human information acquisition: Understanding gatekeeping and proposing new directions in scholarship," *Library and Information Science Research*, vol. 29, pp. 103-123, 2007.

[11] K. S. Rioux, "Information acquiring-and-sharing in Internet-based environments: An exploratory study of individual user behaviors," Ph.D. dissertation, University of Texas at Austin, Austin, TX, 2004.

[12] S. J. Tepper and E. Hargittai, "Pathways to music exploration in a digital age " *Poetics*, vol. 37, pp. 227-249 2009.

[13] R. Sinha and K. Swearingen, "Comparing recommendations made by online systems and friends," in *Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*, 2001.

[14] A. C. North and D. J. Hargreaves, "Music and adolescent identity," *Music Education Research*, vol. 1, pp. 75-92, 1999.

[15] G. H. Lewis, "Who do you love? The dimensions of musical taste," in *Popular music and communication*, J. Lull, Ed., ed Newbury Park, CA: Sage Publications, 1992, pp. 134-151.

[16] S. Frith, *Sound effects : youth, leisure, and the politics of rock'n'roll*. New York: Pantheon Books, 1981.

[17] R. L. Brown and M. O'Leary, "Pop music in an English secondary school system," *American Behavioral Scientist*, vol. 14, pp. 401-413, 1971.

[18] H. Liu, "Social network profiles as taste performances," *Journal of Computer-Mediated Communication*, vol. 13, pp. 252-275, 2007.

[19] S. Wasserman and K. Faust, *Social network analysis: methods and applications*. New York: Cambridge University Press, 1994.

[20] C. Haythornthwaite, "Social Network Analysis: An Approach and Technique for the Study of Information Exchange," *Library and Information Science Research*, vol. 18, pp. 323-342, 1996.

[21] P. Marsden, "Recent developments in network measurement," *Models and methods in social network analysis*, vol. 8, p. 30, 2005.

[22] Y. S. Lincoln and E. G. Guba, *Naturalistic inquiry*. Beverly Hills, CA: Sage Publications, 1985.

[23] R. Curtis, *The future use of social networks in mental health*. Boston: Social Matrix Research, 1979.

[24] M. Granovetter, "The strength of weak ties," *American Journal of Sociology*, vol. 78, p. 1360, 1973.

[25] M. S. Granovetter, "The strength of weak ties: a network theory revisited," *Sociological Theory*, vol. 1, pp. 201-233, 1983.

[26] T. W. Adorno, "On the fetish character in music and the regression of listening," in *The culture industry : selected essays on mass culture*, J. M. Bernstein, Ed., ed London; New York: Routledge, 2001, pp. 29-60.

Content-based music recommendation based on user preference examples

Dmitry Bogdanov, Martín Haro, Ferdinand Fuhrmann, Emilia Gómez, Perfecto Herrera
Music Technology Group
Universitat Pompeu Fabra
Roc Boronat, 138, 08018 Barcelona, Spain
{firstname.lastname}@upf.edu

ABSTRACT

Recommending relevant and novel music to a user is one of the central applied problems in music information research. In the present work we propose three content-based approaches to this task. Starting from an explicit set of music tracks provided by the user as evidence of his/her music preferences, we infer high-level semantic descriptors, covering different musical facets, such as genre, culture, moods, instruments, rhythm, and tempo. On this basis, two of the proposed approaches employ a semantic music similarity measure to generate recommendations. The third approach creates a probabilistic model of the user's preference in the semantic domain. We evaluate these approaches against two recommenders using state-of-the-art timbral features, and two contextual baselines, one exploiting simple genre categories, the other using similarity information obtained from collaborative filtering. We conduct a listening experiment to assess familiarity, liking and further listening intentions for the provided recommendations. According to the obtained results, we found our semantic approaches to outperform the low-level timbral baselines together with the genre-based recommender. Though the proposed approaches could not reach a performance comparable to the involved collaborative filtering system, they yielded acceptable results in terms of successful novel recommendations. We conclude that the proposed semantic approaches are suitable for music discovery especially in the long tail.

Categories and Subject Descriptors

H.3.3 [**Information Storage And Retrieval**]: Information Search and Retrieval—*information filtering, selection process, retrieval models*; H.5.5 [**Information Interfaces And Presentation**]: Sound and Music Computing—*modeling, systems*

General Terms

Algorithms, Measurement, Human Factors

WOMRAD 2010 Workshop on Music Recommendation and Discovery, colocated with ACM RecSys 2010 (Barcelona, SPAIN)

Keywords

recommender systems, user modeling, evaluation, music recommendation, content-based, collaborative filtering

1. INTRODUCTION

Rapid growth of digital technologies, the Internet, and the multimedia industry has provoked a huge information overload and a necessity of effective information filtering systems, and in particular recommendation systems. In the case of the digital music industry, current major Internet stores contain millions of tracks, which complicates search, retrieval, and discovery of music relevant for a user. At present, the majority of industrial systems provide means for contextual manual search based on information about artist names, album or track titles, and additional semantic properties, which are mostly limited to genres. Using this information music collections are becoming browsable by textual queries and tags.

Besides, current research within the music information retrieval (MIR) community achieved relative success in the task of measuring music similarity [7], striving for facilitation of manual search, and automatization of music recommendation. To this extent, music tracks can be represented in a certain feature space filled in with contextual information, extracted from available metadata, user ratings [18], and social tags [12] (i.e. the contextual approach), or with information, extracted from audio content itself [4, 6, 16, 17, 21] (i.e. the content based approach). Thus, it becomes possible to define many similarity measures (or distances[1]) between tracks in a music collections, and therefore to browse collections and to recommend music using queries-by-example. Still the majority of the content-based distances employ solely rough timbral information, such as Mel frequency cepstral coefficients (MFCCs), and sometimes temporal information. Additionally, current systems provide basic means for personalization, obtaining a user's profile in form of consuming statistics, music ratings, and other types of behavioral information, and operating with this information generally in a collaborative filtering manner [2, 8, 9]. While more sophisticated personalization approaches which explore the nature of preference behavior using both contextual information and audio content information are necessary, they are still in their infancy [13–15, 19, 22] and require more research attention.

Generally, we can discern two types of user interaction

[1]We will pragmatically use the term distance to refer to any dissimilarity measurement between tracks.

with a music retrieval system: (i) music search, when a user has an initial idea of what he/she wants, and operates with metadata to query for a specific artist, album, genre, etc., or provides a query-by-example in the case of similarity-based retrieval, and (ii) music discovery, when a user does not know his/her exact needs and prefers to browse an available music collection on purpose to discover music which is relevant in respect to his/her musical preferences. Querying by example requires a user to explicitly define the "direction of search", and is not perfectly suited for discovery. On the other hand, querying by broad semantic categories (such as genres) can provide an excessive amount of potentially relevant data, containing thousands of tracks. While for both types of interaction contextual information can be used, it is found that contextual approaches perform well on popular items, but fail in the long tail due to the lack of available user ratings, social tags, and metadata for unpopular items [8]. Instead, content-based information extracted from audio can help to overcome this problem.

We focus the present work on content-based music recommendation, concerning both relevance and novelty (i.e. discovery) aspects. We do not consider the issue of balancing both aspects according to a user's current needs. Instead, we present a way to infer user preferences from audio content, and a number of recommendation approaches, which are challenged to provide both relevant and novel recommendations to a user. We propose a procedure to generate such recommendations based on an explicit set of music tracks defined by a given user as evidence of his/her musical preferences. Up to our knowledge this recommendation approach has never been evaluated before. We ask the user to provide such a preference set (Sec. 2.1) in order to extract low-level audio features as well as infer high-level semantic information from the audio of each of the tracks (Sec. 2.2). We then consider three different approaches operating on a semantic domain to summarize the retrieved descriptions and generate music recommendations. Two of them have a music similarity measure in their core (Secs. 2.3.1, and 2.3.2), while the third approach applies a probabilistic model to infer the underlying structure of the user's preferences (Sec. 2.3.3). Alternatively, in order to evaluate the generated recommendations, we employ two approaches, which apply the same ideas on low-level timbral features (Secs. 2.3.4, and 2.3.5), and two contextual ones including a state-of-the-art collaborative filtering recommendation system (Sec. 2.3.6), and a naive genre-based recommender baseline (Sec. 2.3.7). We evaluate all considered approaches by gathering music data from 12 participants (Sec. 3.1), and carrying out a listening experiment to assess familiarity, liking and further listening intentions of the provided recommendations (Sec. 3.2), and present the obtained results (Sec. 3.3). Finally, we draw conclusions about the proposed procedure and discuss future research directions (Sec. 4).

2. METHODOLOGY

2.1 Preference Examples Collection

As a first step, we ask the user to gather the minimal set of music tracks sufficient to grasp or convey her/his music preferences [10] (the user's preference set). We do not promise or mention giving music recommendations in the future, which could bias the selection of representative music. The user provides a folder with the selected tracks in audio format (e.g. mp3), and all the needed information to unambiguously identify and retrieve each track (i.e. artist, piece title, edition, etc.). For the content-based approaches which we will consider, single music pieces are informative by themselves without any additional context, such as artist names and track titles. Still we ask the user to provide this context to be able to make comparison with contextual approaches. We also ask the user for additional information, including personal data (gender, age, interest for music, musical background), a description of the strategy followed to select the music pieces, and the way he/she would describe his/her musical preferences. This information will help us for further analysis.

2.2 Audio Content Analysis

We now describe the procedure of obtaining meaningful low-level and high-level descriptions of each music track from the user's preference set within the used audio content analysis system. We follow [6] to obtain such descriptions. To this extent, for each track we calculate a low-level feature representation using an in-house audio analysis tool[2]. In total it provides over 60 commonly used low-level audio features, characterizing global properties of the given tracks, including timbral, temporal, and tonal features among others. They include inharmonicity, odd-to-even harmonic energy ratio, tristimuli, spectral centroid, spread, skewness, kurtosis, decrease, flatness, crest, and roll-off factors, MFCCs, spectral energy bands, zero-crossing rate, spectral and tonal complexities, transposed and untransposed harmonic pitch class profiles, key strength, tuning, chords, beats per minute and onsets.

We do not use the described low-level features explicitly in the approaches we will consider, except for MFCCs, used to construct two of the baseline systems. Instead, we use them to infer semantic descriptors. For that reason, we perform a regression by suitably trained classifiers producing different semantic dimensions such as genre, culture, moods, and instrumentation. We use standard multi-class support vector machines (SVMs) [20], employ 14 ground truth music collections (including full tracks and excerpts) and execute 14 classification tasks corresponding to these data. The regression results form a high-level descriptor space, which contains the probability estimates for each class of each SVM classifier. With the described procedure we obtain 56 high-level descriptors, including categories of genre, culture, moods, instruments, rhythm and tempo. For more detailed information regarding the list of low-level features, the collections used for regression, and SVM implementation see [6] and references therein.

2.3 Recommendation Approaches

We now consider different approaches to music recommendation, which are based on the retrieved descriptions of the user's preference set. The approaches we propose include three methods working on semantic descriptors. In comparison, we consider two low-level baseline approaches working on MFCCs, and two contextual ones.

All approaches are used to retrieve 20 music tracks from a given music collection as the recommendations for the user except one of the contextual approaches (Sec. 2.3.6), which operates on *Last.fm*[3] music collection.

[2] `http://mtg.upf.edu/technologies/essentia`
[3] `http://last.fm`

2.3.1 Semantic distance from the mean (SEM-MEAN)

As the simplest approach, we propose the representation of the user as a single point in the semantic descriptor space. As such, we compute the mean point for the user's preference set. We employ the semantic distance, presented and validated in [6]. It has been shown to perform with positive user satisfaction, being comparable to well-known low-level timbral distances, based on MFCCs, while operating in a high-level semantic space. More concretely, the distance operates directly on the retrieved semantic descriptors, and is defined as a weighted Pearson correlation distance [1, 6]. Given a music collection, we rank the tracks according to the semantic distance to the user point (i.e. the mean point of the user's preference set) and return 20 nearest tracks as recommendations.

2.3.2 Semantic distance from all tracks (SEM-ALL)

Alternatively, we do not simplify the user representation to one point but instead consider all tracks from the user's preference set. We use the same semantic distance as for SEM-MEAN. For each track from the user's preference set, we compute the distances to the tracks in a given music collection, and mark 20 nearest tracks as candidates. We then rank all selected candidates according to the obtained distances, omit possible duplicates, and return the tracks corresponding to the lowest 20 distances as recommendations. In this case, we take into account all possible areas of preferences, explicitly specified by the user, while searching for the most similar tracks.

2.3.3 Semantic Gaussian mixture model (SEM-GMM)

Finally, we propose the representation of the user as a probability density of his/her preferences on the semantic space. For that purpose, we use the retrieved semantic descriptors, and employ a Gaussian mixture model (GMM) [5], which estimates a probability density as a weighted sum of a given number of simple Gaussian densities (components). We initialize the GMM by k-mean clustering, and train the model using the expectation-maximization algorithm. The number of centers for the k-means are estimated by Bayesian information criterion [5]. For computational reasons, we consider a number of components in the range between 1 and 20. Once we have our model trained, we compute probability density for each of the tracks in a given music collection. We rank the tracks according to the obtained density values, and return 20 most probable tracks as recommendations under the assumption of a uniform distribution of the tracks in the universe within the semantic space.

The advantage of SEM-GMM approach is that the model takes the relevance of the semantic attributes within the user's preferences into account, accenting areas preferred by the user in the semantic space. Thus, the recommended tracks would generally comprise of the most characteristic semantic properties, inferred from the user's preference set. Meanwhile, SEM-ALL is blind to the underlying semantic structure of preferences, and SEM-MEAN only provides very rough approximation. Still, in the case when the user's tracks are evenly distributed in the semantic space, SEM-GMM may have insufficient expressive power due to the assigned limit of Gaussian components, discriminating certain preference areas. Nonetheless we assume gaussianity of the user's preference patterns.

2.3.4 Timbral distance from all tracks (MFCC-ALL)

For comparison purposes and as our first baseline we modify the SEM-ALL approach to use a common low-level timbral distance [16] instead of the semantic one. To this extent, we use MFCCs and model each music track as a single Gaussian with full covariance matrix. A closed form symmetric approximation of the Kullback-Leibler divergence is then used as a distance. Thereby, we can regard the MFCC-ALL approach as a counterpart of the distance-based approach to music recommendation proposed by Logan [14] in which the Earth-Mover's Distance between MFCC clusters is used as a distance measure.

2.3.5 Timbral Gaussian mixture model (MFCC-GMM)

Alternatively, as in the SEM-GMM approach, we construct a probabilistic model using a GMM. Instead of the semantic descriptors, we use a population of mean MFCC vectors (one vector per track) to train the model.

2.3.6 Collaborative filtering with Last.fm (LASTFM)

In addition to the described content-based approaches, we consider a contextual baseline approach based on music similarity inferred from collaborative filtering information. We did not have at hand any data of this kind on our own, and therefore we opted for the usage of black box recommendations, provided by *Last.fm*. It is an established music recommender with an extensive number of users, and a large music collection, providing means for both monitoring listening statistics and social tagging [11].

We manually generate a list of recommendations browsing *Last.fm*. The procedure we follow for that purpose partially emulates human user behavior while discovering new music. During the retrieval procedure we did not open any account for *Last.fm*, therefore we consider such recommendations unbiased to possible personalization, which can be provided for the registered accounts. We randomly preselect 20 music tracks from the user's preference set, and query the *Last.fm* website for each of the preselected tracks. To this extent, for each query track, we search a corresponding *Last.fm* track page[4]. If the track page is found, we pass to the "Similar Music" page[5]. This page provides a ranked list of tracks similar to the query track. From the list we select the first track which is available for pre-listen online, by a different artist than the query track. Otherwise, if the corresponding track page is not found, or the "Similar Music" page is not available for the query track due to insufficient collaborative filtering data (e.g., when the query track is an unpopular long-tail track with low number of listeners), we search for the corresponding artist page[6] and proceed to the "Similar Artists" page[7]. This page provides a ranked list of artists, similar to the artist of the query track. We apply an artist filter to the list as the query artist name can have variations. Thereafter we select the top-ranked artist from the list, go to the corresponding artist page, and select the first track, which is available for pre-listen online, from the "Top Tracks" section. This section provides two lists of the most popular tracks by the artist, relying on short-term last

[4] for example, see `http://www.last.fm/music/Mastodon/_/The+Czar`

[5] `http://www.last.fm/music/Mastodon/_/The+Czar/+similar`

[6] `http://www.last.fm/music/Baby+Ford/`

[7] `http://www.last.fm/music/Baby+Ford/+similar`

week period, or long-term last 6 months period of listening statistics. We opted for the last 6 months period. If no prelistens are found, we proceed iteratively to the next similar artist's top tracks, until we find one. If no similar artist contains previewable tracks, we skip the query track.

2.3.7 Random tracks by the same genre (GENRE)

Finally, as a simple and low-cost contextual baseline, we provide random recommendations, which rely on genre categories of the user's tracks. As in the LASTFM approach, we preselect 20 music tracks from the user's preference set. For each of the tracks we obtain a genre category of this track from the *Last.fm* track page, or artist page. As such, we select the first genre tag we encounter, which is presented in a given music collection (we assume, that all tracks are tagged with a genre category). Thereafter, we return a random track of this genre tag from the collection.

3. EXPERIMENTS AND RESULTS

3.1 User Data Analysis

We worked with a group of 12 users (8 males and 4 females). They were aged between 25 and 45 years old (average $\mu = 32.75$ years old and standard deviation $\sigma = 5.17$ years old) and showed a very high interest in music (rating around $\mu = 9.58$, with $\sigma = 0.67$, where 0 means no interest in music and 10 means passionate about music). Ten of the twelve users play at least one musical instrument, including violin, piano, guitar, singing, synthesizers and ukulele.

The number of tracks selected by the users to convey their musical preferences was very varied, ranging from 19 to 178 music pieces ($\mu = 73.25$, $\sigma = 46.07$). The time spent for this task also differed a lot, ranging from half an hour to 180 hours ($\mu = 30.41$, $\sigma = 54.19$).

It is interesting to analyze the provided verbal descriptions about the strategy followed to select the music tracks. Some of the users were selecting one song per artist, while some others did not apply this restriction. They also covered various uses of music such as listening, playing, singing or dancing. Other users mentioned musical genre, mood, expressivity, musical parameters, lyrics and chronological order as driving parameters for selecting the tracks. Furthermore, some users implemented an iterative strategy by gathering a very large amount of music pieces from their music collection and performing a further refinement to obtain the final selection.

Finally, each user provided a set of labels to define their musical preferences. Most of them were related to genre, mood and instrumentation, some of them to rhythm and few to melody, harmony or expressivity. Other labels were attached to lyrics, year and duration of the piece. The users' preferences covered a wide range of musical styles (from classical to country, jazz, rock, pop, electronic, folk) and musical properties (e.g. acoustic vs. synthetic, calm vs. danceable, tonal and dissonant).

3.2 Recommendation Evaluation

In order to evaluate the considered approaches, we performed subjective listening tests on our 12 subjects. The entire process used an in-house collection of 100K music excerpts (30 sec.) by 47K artists (approximately 2 tracks per artist) covering a wide range of musical dimensions (different genres, styles, arrangements, geographic locations, and epochs). For each user we generated 7 recommendation playlists, using each of the three proposed approaches and two low-level plus two contextual baseline approaches. Each playlist consisted of 20 music tracks, returned by the respective approach specifics (Sec. 2.3). No playlist contained more than one song from the same artist. All playlists were merged into a single list of 140 tracks, with all the tracks randomly ordered to avoid any response bias because of presentation order or because of recommendation approach. The file names were anonymized, and all metadata was deleted from the files as well, to make contextual identification of the tracks impossible. Also the participants were not aware of the amount of recommendation approaches, their names and their rationales.

Table 1: The percent of fail, trust, hit, and unclear categories per recommendation approach. Note that the results for the LASTFM approach were obtained on a different underlying music collection.

Approach	fail	hit	trust	unclear
SEM-MEAN	49.167	31.250	2.500	17.083
SEM-ALL	42.500	34.583	3.333	19.583
SEM-GMM	48.750	30.000	2.500	18.750
MFCC-ALL	64.167	15.000	2.083	18.750
MFCC-GMM	69.583	11.667	1.250	17.500
LASTFM	16.667	41.250	25.417	16.667
GENRE	53.750	25.000	1.250	20.000

A questionnaire was given for the subjects to express different subjective impressions related to the recommended music. A "familiarity" rating ranged from the identification of artist and title (4) to absolute unfamiliarity (0), with intermediate steps for knowing the title (3), the artist (2), or just feeling familiar with the music (1). A "liking" rating measured the enjoyment of the presented music with 0 and 1 covering negative liking, 2 being a kind of neutral position, and 3 and 4 representing increasing liking for the musical excerpt. A rating of "listening intentions" measured preference, but in a more direct and behavioral way than the "liking" scale, as an intention is closer to action than just the abstraction of liking. Again this scale contained 2 positive and 2 negative steps plus a neutral one. Finally, an even more direct rating was included with the name "gimmemore" allowing just 1 or 0 to respectively indicate a request for, or a reject of, more music like the one presented. The users were also asked to provide title and artist for those tracks rated high in the familiarity scale. We manually corrected this scale when the given artist/title was wrong (hence a familiarity rating of "3" or, more frequently, "4", was sometimes lowered to 1. These corrections represented just 3% of the total familiarity judgments.

3.3 Results

Considering the subjective scales used, a good recommendation system should provide high-liking/listening intentions/request for the greater part of retrieved tracks and in particular for low-familiarity tracks. Therefore, we recoded the user's ratings into 3 main categories, referring to the type of the recommendation: hits, fails and trusts. Hits were those tracks having a low familiarity rating (< 2) and a high (> 2) liking rate. Fails were those tracks having a low (< 3) liking rating. Trusts were those tracks that got a

high familiarity (> 1) and a high (> 2) liking rate. Trusts, provided their overall amount is low, can be useful for a user to feel that the recommender is understanding his/her preferences [3] (i.e., a user could be satisfied by getting a trust track from time to time, but annoyed if every other track is a trust). Using the liking, the intentions and the "gimmemore" Boolean rating we respectively computed three different recommendation outcome measures. Then we combined the three into a final recommendation outcome that required absolute coincidence of them in order to consider it to be a hit, a fail or a trust. A 18.3% of all the recommendations were then considered as "unclear" (e.g., a case that, using the liking, it was a hit, but using the other two indexes it was a fail), and were excluded from further analyzes. An interesting additional result is that many of the unclear outcomes correspond to high-liking ratings that turned into 0 in the gimmemore scale. This pattern was more frequent for the recommendations generated using the GMM-MFCC (6.6%) than for any other approaches, being the GENRE the least changed (2.9%). Contrastingly, the opposite change (low-liking becoming positive "gimmemore") was nearly absent in the ratings.

The percent of each category per recommendation approach is presented in Table 1. An inspection of it reveals that the approach yielding more hits (41.2%) and trusts (25.4%) is LASTFM (not surprisingly the trusts found with other approaches were scarce, below 4%). The three approaches based on semantic descriptors (SEM-ALL, SEM-MEAN and SEM-GMM) yielded more than 30% of hits, and the remaining ones could not supply more than 25%. The existence of an association between recommendation approach and the outcome of the recommendation could be accepted, according to the result of the Pearson chi-square test ($\chi^2(18) = 351.7$, $p < 0.001$).

Additionally, three separate between-subjects ANOVA were performed in order to test the effects of the recommendation approaches on the three subjective ratings. The effect was confirmed in all of them ($F(6, 1365) = 55.385$, $p < 0.001$ for the liking rating, $F(6, 1365) = 48.89$, $p < 0.001$ for the intentions rating, and $F(6, 1365) = 43.501$, $p < 0.001$ for the "gimmemore" rating). Pairwise comparisons using Tukey's test revealed the same pattern of differences between the recommendation approaches, irrespective of the 3 tested indexes. This pattern highlights the LASTFM approach as the one getting the highest overall ratings, it also groups together the MFCC-GMM and MFCC-ALL approaches (those getting the lowest ratings), and the remaining approaches also clustered in-between.

Finally, a measure of the quality of the hits was computed doing (liking $-$ familiarity) $*$ intentions. Selecting only the hits, an ANOVA on the effect of recommendation method on this quality measure revealed no significant differences attributable to the method. Therefore, once a hit is selected, there is no recommendation method granting better or worst recommendations than any other. The same pattern was revealed by solely using the liking as a measure of the quality of the hits.

4. CONCLUSIONS

In this work we presented three content-based approaches to music recommendation, which are based on an explicit set of music tracks provided by a user as evidence of his/her musical preferences (the user's preference set). Our approaches work on semantic descriptors (inferred from low-level audio features in diverse classification tasks) covering musical dimensions such as genre and culture, moods and instruments, and rhythm and tempo. More concretely, we proposed two approaches which apply a high-level semantic distance to retrieve tracks from a given collection. These approaches compute the distance either from the mean point of the preference set, or from all tracks in the preference set. Alternatively, we proposed a model-based approach, which creates a probabilistic model to infer the underlying structure of the user's preferences. For that purpose, we employed a GMM to model the preferences within the semantic domain. We evaluated the proposed approaches against a number of baselines in a subjective evaluation with 12 users. As such baselines, we considered two approaches operating on low-level timbral features (MFCCs) instead of the proposed semantic descriptors. The first approach employs a state-of-the-art timbral distance, while the second one creates a GMM within the timbral domain. Moreover, in contrast to the content-based methods, we included two contextual recommenders in our evaluation. One of them naively retrieves random tracks from a given music collection by a genre criterion. The other employs *Last.fm* as a source for collaborative filtering information about music similarity.

The evaluation results revealed the user's preference of the proposed semantic approaches over the low-level timbral baselines. This concerns both the compared distance-based approaches as well as the probabilistic models. Regarding the semantic distance employed in our approaches, this fact supports and complements the outcomes from the previous research on semantic music similarity measures [6], in which a number of similarity measures were evaluated in a subjective experiment but on a set of random tracks not necessarily preferred by participants. In that experiment a comparable performance of the semantic and low-level timbral distances was revealed, meanwhile the semantic distance surpassed the other methods in objective evaluations. Considering these previous results and the present outcomes, we may conclude that the high-level semantic description outperforms the low-level timbral description in the task of music recommendation.

In contrast, the proposed approaches are found to be inferior to the considered collaborative filtering recommender in terms of both the number of successful novel recommendations (hits) and the trusted recommendations. This result can be partly explained by the fact that the recommendations generated by the latter approach used the *Last.fm* music collection, which could entail an evaluation bias. Considering this fact, we can hypothesize a lower performance of the collaborative filtering approach on our in-house collection. Still the collaborative filtering approach yielded only 7% more hits than our best proposed semantic method. In particular, we 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.

Interestingly, the naive genre-based recommender, while being worse than our proposed approaches, still outperformed the timbre-based baselines. This could be partially explained by the fact that genre was one of the driving criteria for selecting users' preference sets, and that genre entails more information and diversity than timbral information extracted from MFCCs. We also did not find benefits of using our semantic GMM-based approach comparing to the semantic

distance-based approaches, probably due to the insufficient size of training data (only one mean MFCC vector per track was computed in our experiments).

In general, we conclude that though the considered content-based approaches to music recommendation do not reach the satisfaction and novelty degree of the collaborative filtering approach, the difference in performance diminishes to a great extent while using semantic descriptors. We may hypothesize a better performance, comparable with the collaborative filtering approach, once the amount and quality of semantic descriptors is increased. Consequently, future research will be devoted to the extension of the inherent semantic descriptor space, used by the proposed approaches, as well as the improvement of the underlying classifiers, and the distance measure. Furthermore, we plan to assess the potential benefit of user profiling by explicitly given preference examples in form of music tracks over more broad contextual categories (favorite artists, albums, genres, and even activities), and implicit information such as listening behavior statistics.

5. ACKNOWLEDGMENTS

The authors would like to thank all participants involved in the evaluation. This research has been partially funded by the FI Grant of Generalitat de Catalunya (AGAUR).

6. REFERENCES

[1] M. B. Abdullah. On a robust correlation coefficient. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 39(4):455–460, 1990.

[2] L. Baltrunas and X. Amatriain. Towards time-dependant recommendation based on implicit feedback. In *Workshop on Context-aware Recommender Systems (CARS'09)*, 2009.

[3] L. Barrington, R. Oda, and G. Lanckriet. Smarter then genius? human evaluation of music recommender systems. In *10th International Society for Music Information Retrieval Conference (ISMIR'09)*, 2009.

[4] L. Barrington, D. Turnbull, D. Torres, and G. Lanckriet. Semantic similarity for music retrieval. In *International Symposium on Music Information Retrieval (ISMIR'07)*, 2007.

[5] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[6] D. Bogdanov, J. Serrà, N. Wack, and P. Herrera. From low-level to high-level: Comparative study of music similarity measures. In *International Workshop on Advances in Music Information Research (AdMIRe'09)*, 2009.

[7] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

[8] O. Celma. *Music recommendation and discovery in the long tail*. PhD thesis, UPF, Barcelona, Spain, 2008.

[9] C. S. Firan, W. Nejdl, and R. Paiu. The benefit of using tag-based profiles. In *Latin American Web Conference*, pages 32–41, 2007.

[10] M. Haro, A. Xambó, F. Fuhrmann, D. Bogdanov, E. Gómez, and P. Herrera. The musical avatar - a visualization of musical preferences by means of audio content description. In *Audio Mostly (AM '10)*, Pitea, Sweden, 2010. ACM.

[11] N. Jones and P. Pu. User technology adoption issues in recommender systems. In *Networking and Electronic Commerce Research Conference*, 2007.

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

[13] Q. Li, S. H. Myaeng, and B. M. Kim. A probabilistic music recommender considering user opinions and audio features. *Information Processing & Management*, 43(2):473–487, Mar. 2007.

[14] B. Logan. Music recommendation from song sets. In *Proc ISMIR*, page 425–428, 2004.

[15] C. C. Lu and V. S. Tseng. A novel method for personalized music recommendation. *Expert Systems with Applications*, 36(6):10035–10044, 2009.

[16] E. Pampalk. *Computational models of music similarity and their application in music information retrieval*. PhD thesis, Vienna University of Technology, Mar. 2006.

[17] T. Pohle, D. Schnitzer, M. Schedl, P. Knees, and G. Widmer. On rhythm and general music similarity. In *10th International Society for Music Information Retrieval Conference (ISMIR'09)*, 2009.

[18] M. Slaney and W. White. Similarity based on rating data. In *International Symposium on Music Information Retrieval (ISMIR'07)*, 2007.

[19] J. Su, H. Yeh, P. S. Yu, and V. S. Tseng. Music recommendation using content and context information mining. *IEEE Intelligent Systems*, 25(1):16–26, 2010.

[20] V. Vapnik. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer, 2nd edition, Nov. 1999. Published: Hardcover.

[21] K. West and P. Lamere. A model-based approach to constructing music similarity functions. *EURASIP Journal on Advances in Signal Processing*, 2007:149–149, 2007.

[22] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In *International Conference on Music Information Retrieval (ISMIR'06)*, 2006.

Applying Constrained Clustering for Active Exploration of Music Collections

Pedro Mercado
Fraunhofer Institute for Digital Media Technology
Ehrenbergstr. 31
Ilmenau, Germany
mercpo@idmt.fraunhofer.de

Hanna Lukashevich
Fraunhofer Institute for Digital Media Technology
Ehrenbergstr. 31
Ilmenau, Germany
lkh@idmt.fraunhofer.de

Instituto Tecnológico Autónomo de México
Río Hondo No. 1, Tizapán, 64230
DF, México
pmercadol@comunidad.itam.mx

ABSTRACT

In this paper we investigate the capabilities of constrained clustering in application to active exploration of music collections. Constrained clustering has been developed to improve clustering methods through pairwise constraints. Although these constraints are received as queries from a noiseless oracle, most of the methods involve a random procedure stage to decide which elements are presented to the oracle. In this work we apply spectral clustering with constraints to a music dataset, where the queries for constraints are selected in a deterministic way through outlier identification perspective. We simulate the constraints through the ground-truth music genre labels. The results show that constrained clustering with deterministic outlier identification method achieves reasonable and stable results through the increment of the number of constraint queries.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information Storage and Retrieval; H.5.5 [**Information Systems**]: Sound and Music Computing—*methodologies and techniques*

General Terms

Theory, Experimentation, Algorithms

Keywords

Constrained clustering, outlier identification, spectral clustering, active semi-supervised learning, music information retrieval

WOMRAD 2010 Workshop on Music Recommendation and Discovery, colocated with ACM RecSys 2010 (Barcelona, SPAIN)

1. INTRODUCTION

During recent years the scientific and commercial interest in Music Information Retrieval (MIR) has significantly increased. Stimulated by the ever-growing availability and the size of digital music collections, automatic music indexing and retrieval systems has been identified as an increasingly important means to aid convenient exploration of large music catalogs. In order to supply the users with more accurate and robust music exploration systems, automatically extracted metadata like "music genre", "style" or "mood" can be added to the conventional metadata e.g. artist name, album name and track title. Commonly this automatically extracted metadata is derived by means of collaborative filtering or is generated by statistical classifiers that are pretrained on the restricted amount of labeled ground-truth data. The exploration intentions of the end-user might not be expressed by the available training data. Hence the desirable exploration facets might stay unreachable.

An alternative way of music exploration is to visualize the music collection or a part of it by placing similar songs close to each other and non-similar songs far away from each other in some low-dimensional space projection. A comprehensive overview of the existing up to date systems and methods can be found in [18]. Similar goals can also be reached with clustering algorithms that cluster (group) songs in a way that similar songs are joined in clusters and non-similar songs appear in different clusters. Obviously, music has too many facets (aspects) for one "static" clustering that allows to use only one definition of similarity. In this paper we consider clustering with constraints as a complimentary fashion to music collection exploration. Here the user can express a particular point to clusterability of his/her music collection by providing some feedback information in the form of constraints. Clustering with constraints has been already applied to a music collection by Peng et al. [15]. They simulated the generation of constraints by choosing random constraint pairs from the classes in artist similarity graph. We propose to avoid using random constraints. In contrast, we determine the optimal songs to be constrained via outlier identification methods.

The reminder of the paper is organized as follows. Sec. 2 provides some theoretic background on applied clustering and outlier identification methods. The conception of the conducted experiments is presented in Sec. 3. In Sec. 4 we bring some details on audio features, utilized dataset, evaluation scenarios and evaluation measures. The results are presented and discussed in Sec. 5 and Sec. 6 concludes the paper and brings some insights to the future research directions.

2. MATHEMATICAL BACKGROUND

In this section we present the fundamental concepts and methods used in this paper. We will always consider a data set X of n elements such that $X = \{\mathrm{x_1}, \mathrm{x_2}, \ldots, \mathrm{x_n}\}$, where $\mathrm{x_i} \in \mathbb{R}^m$. See Sec. 4 for details on the used dataset.

2.1 Graph Laplacian

The fundamental tool related to spectral methods is the graph Laplacian. We present it briefly.

Let $S \in \mathbb{R}^n$ be a similarity matrix related to dataset X, $G = (E, V)$ a similarity graph where E and V are the sets of edges and vertexes, respectively and W its corresponding weighted adjacency matrix. Let D be the degree matrix, which has $d_{ii} = \sum_{j=1}^{n} w_{ij}$ and zero elsewhere. Then, the unnormalized (L), Symmetric (L_{sym}) and Random Walk (L_{rw}) Laplacians are:

$$\begin{aligned} L &= D - W \\ L_{sym} &= D^{-1/2} L D^{-1/2} \\ L_{rw} &= D^{-1} L \ . \end{aligned} \tag{1}$$

Some of the properties that the Laplacians hold are:

1. They are symmetric positive semi-definite.
2. They have n real non-negative eigenvalues
3. The multiplicity of the smallest eigenvalue, which is always zero, is equal to the number of connected components of G

2.2 Spectral Feature Selection

The properties of Laplacian operators have been already extended to feature selection methods. In particular Zhao et al. [22] have developed a filter method based on properties of Laplacians. We present it briefly.

Given a graph G, its corresponding weighted adjacency matrix W and degree matrix D, let λ_j and ξ_j be the eigenvalues and eigenvectors of the corresponding symmetric Laplacian L_{sym} with $0 \leq \lambda_1 \leq \cdots \leq \lambda_n$. Then the score of the feature F_i can be measured through the following function:

$$\varphi(F_i) = \sum_{j=2}^{k} (\gamma(2) - \gamma(\lambda_j))\alpha_j^2 \ , \tag{2}$$

where γ is a rational function, k is a number of clusters and α_j is a cosine of the angle between the eigenvector ξ_j and the weighted feature $\widehat{\boldsymbol{f}}_i$ which is defined as

$$\widehat{\boldsymbol{f}}_i \ = \ \frac{D^{1/2}\boldsymbol{f}_i}{\parallel D^{1/2}\boldsymbol{f}_i \parallel} \ , \tag{3}$$

where $\boldsymbol{f}_i$ is the feature vector corresponding to F_i.

Score function in eq. (2) considers the same criteria as spectral clustering, where the first k eigenvectors are the most relevant ones. This function assigns high values to features which give better separability for a given number of clusters in the graph G. Therefore, features should be ranked in descending ordered through the given feature score.

2.3 Clustering Methods

In this part we present two fundamental approaches considered in this paper: constrained clustering and spectral clustering.

2.3.1 Constrained Clustering

It is not always possible to get true labels, even for just a portion of a dataset. In some circumstances it may be possible to get information between pairs of elements. Wagstaff et al. [21] proposed the addition of information through pairwise constraints. They introduced two types of pairwise constraints: namely Must Links (ML) if two elements should be in the same cluster, and Cannot Links (CL) if two elements should be in different clusters. This fundamental idea has been already applied for center initialization through weighted farthest traversal heuristic by Basu et al. [4] and even generalized to kernel and graph methods by Kulis et al. [11]. In particular, they exposed the manner in which the information from given constraints can be added to this clustering methods. Given an affinity matrix W and sets of ML and CL, we define T as the constraint matrix, where for each pair of points $(\mathrm{x_i}, \mathrm{x_j})$

$$T = \{t_{ij}\} : t_{ij} = \begin{cases} m_{ij}, & \text{for a ML,} \\ -m_{ij}, & \text{for a CL,} \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where each m_{ij} is an arbitrary scalar. Then, the matrix which summarizes the side information is

$$W' = W + T \tag{5}$$

and can be used for both kernel and graph clustering methods.

2.3.2 Spectral clustering

Spectral clustering has received a considerable amount of attention, due to its surprising results and easy implementation. We present the general framework related to Random Walk and Symmetric Laplacians. For more details, we refer to Luxburg [14].

Let G and W be respectively the similarity graph and its weighted adjacency matrix obtained from a given similarity matrix. Depending on the type of Laplacian, the Matrix U is obtained as following:

- for *Random Walk Laplacian* we get the first k generalized eigenvectors $u_1, \ldots, u_k$ from the generalized eigenvalue problem $Lu = \lambda Du$ and store them column-wise in a matrix $U \in \mathbb{R}^{n \times k}$.
- for *Symmetric Laplacian* we get the first k eigenvectors $u_1, \ldots, u_k$ of L_{sym}, store them column-wise in a matrix $U \in \mathbb{R}^{n \times k}$ and normalize each row of U.

Afterwards, k-means algorithm is applied to cluster the rows of matrix U, where each row is the embedding of the elements of the given dataset.

2.4 Outlier Identification Methods

Application of the outlier identification is motivated by the intrinsic nature of music, that is in some sense full of outliers. Clustering constrained on extremes rather than "randoms", covers more of the problematic pieces. In this study we apply the following outlier identification methods:

LOF Local Outlier Factor (LOF) was proposed by Breunig et al. [6]. It can be interpreted as an *outlierness degree* and gives the possibility to rank the items through it. As the name suggests, the outlierness of each element is restricted to local neighborhoods.

RRS Ramaswamy et al. [16] considered that the distance of each point to its k^{th} nearest neighbor determines if it is an outlier or not. Hence, the larger the distance, the more chances for the item to be an outlier. Further we address this outlier detection method as RRS.

Both methods provide the possibility to rank the items through their outlier degree. This allows to choose the order in which the elements will be exposed to be constrained.

3. CONCEPTION OF EXPERIMENTS

In this section we explain the integration of the exposed concepts and the setup of the experiments. The process steps described in this section are summarized in Figure 1.

Given a *data set* $\mathbf{X}$ and a set of pairwise *constraints* in all experiments we aim to get the *cluster assignments*. Each item in the dataset is represented with a feature vector $\mathbf{x_i}$, $i = 1, \ldots, n$, where n is a number of elements in the dataset. Not all dimensions in $\mathbf{x_i}$ are equally profitable for the similarity relations between the items in the dataset. In order to select the most appropriate feature dimensions, we apply a *spectral feature selection* method as stated in Sec. 2.2. In our experiments the rational function in eq. (2) is set to $\gamma(x) = x^4$. Given a data set related to the selected features, the similarity relations between the items are captured via the *correlation coefficient kernel* K, where $K(\mathbf{x_i}, \mathbf{x_j})$ is equal to the Pearson correlation coefficient between vectors $\mathbf{x_i}$ and $\mathbf{x_j}$ as

$$K(\mathbf{x_i}, \mathbf{x_j}) = \frac{\sum_k (\mathbf{x_{i,k}} - \bar{\mathbf{x}}_\mathbf{i})(\mathbf{x_{j,k}} - \bar{\mathbf{x}}_\mathbf{j})}{\sqrt{\sum_k (\mathbf{x_{i,k}} - \bar{\mathbf{x}}_\mathbf{i})^2} \sqrt{\sum_k (\mathbf{x_{j,k}} - \bar{\mathbf{x}}_\mathbf{j})^2}} \ , \qquad (6)$$

where $\bar{\mathbf{x}}_\mathbf{i}$ and $\bar{\mathbf{x}}_\mathbf{j}$ are the empirical means of vectors $\mathbf{x_i}$ and $\mathbf{x_j}$ respectively. The matrix related to the kernel is symmetric positive semi-definite.

The correlation coefficient kernel K is utilized to determine the *K Nearest Neighbors matrix* (KNN), where indeed the neighborhood of each song is conformed by the K most correlated songs. Here the parameter K was chosen as $K = log_2(n)$, where n is the number of elements (songs) in the dataset. In addition to the KNN matrix we calculate the *Symmetric K Nearest Neighbors matrix* (SKNN), where the KNN matrix is symmetrized through the insertion of missing non-mutual neighbor connections. The KNN matrix is utilized by the *outlier detection* methods introduced in Sec. 2.4. At this stage we also consider the possibility of getting outliers random-wise just for the sake of comparison of traditional presented scores in the literature.

For a set of identified outliers we get constraints from a noiseless oracle and through the corresponding extended Must-Link sets. The corresponding *weighted adjacency matrix* is defined as

$$W = SKNN + T \ , \qquad (7)$$

where T is the corresponding constraint matrix pointed in eq. (4). Here the elements t_{ij} of the constraint matrix T are set to the maximal (out of main diagonal) value of adjacency matrix W for ML, and to $t_{ij} = -w_{ij}$ for CL. Next, we use either the Symmetric or Random Walk *Laplacian* (see eq. (1)) and apply *spectral clustering* (see Sec. 2.3.2), receiving cluster assignments as outputs.

For our work we consider the following six experiments where outlier identification methods and particular Laplacians are combined as presented in Table 1.

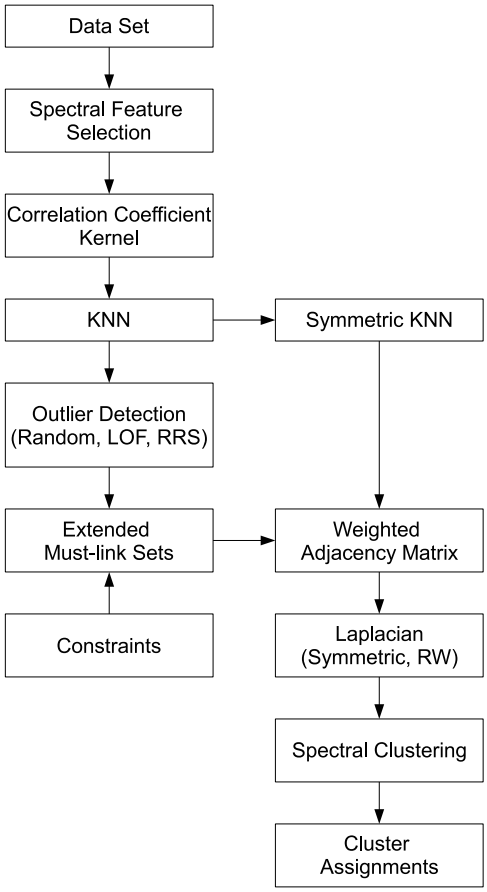


Figure 1: Flow chart diagram of experiments (see Sec. 3 for details)

Table 1: Configuration of experiments

Short name	Laplacian	Outlier Identification
Sym RAW	Symmetric	Random
Sym LOF	Symmetric	LOF
Sym RSS	Symmetric	RRS
RW RAW	Random Walk	Random
RW LOF	Random Walk	LOF
RW RSS	Random Walk	RRS

4. EVALUATION SETUP

In this section we provide some details on the evaluation setup. First of all, we briefly introduce audio features used for compact and informative representation of audio tracks. Afterwards, we describe musical dataset involved in the experiments. Finally, we bring some insights to the evaluation scenarios and the evaluation measures used to estimate the effectiveness of proposed clustering algorithms.

4.1 Audio Features

We utilize a broad palette of low-level acoustic features and several mid-level representations [5]. These mid-level features are computed on 5.12 seconds excerpts and observe the evolution of the low-level features. With the help of mid-level representations, timbre texture [19] can be captured by descriptive statistics as well as by including additional musical knowledge. To facilitate an overview the audio feature are subdivided in three categories by covering the timbral, rhythmic and tonal aspects of sound.

Although the concept of *timbre* is still not clearly defined with respect to music signals, it proved to be very useful for automatic music signal classification. To capture timbral information, we use Mel-Frequency Cepstral Coefficients, Spectral Crest Factor, Audio Spectrum Centroid, Spectral Flatness Measurement, and Zero-Crossing Rate. In addition, modulation spectral features [1] are extracted from the aforementioned features to capture their short term dynamics. We applied a cepstral low-pass filtering to the modulation coefficients to reduce their dimensionality and to decorrelate them as described in [7].

All *rhythmic* features used in the current setup are derived from the energy slope in excerpts of the different frequency-bands of the Audio Spectrum Envelope feature. These comprise the Percussiveness [20] and the Envelope Cross-Correlation (ECC). Further mid-level features [7] are derived from the Auto-Correlation Function (ACF). In the ACF, rhythmic periodicities are emphasized and phase differences annulled. Thus, we compute also the ACF Cross-Correlation (ACFCC). The difference to ECC again captures useful information about the phase differences between the different rhythmic pulses. In addition, the log-lag ACF and its descriptive statistics are extracted according to [10].

Tonality descriptors are computed from a Chromagram based on Enhanced Pitch Class Profiles (EPCP) [12], [17]. The EPCP undergoes a statistical tuning estimation and correction to account for tunings deviating from the equal tempered scale. Pitch-space representations as described in [8] are derived from the Chromagram as mid-level features. Their usefulness for audio description has been shown in [9].

Clustering music tracks that are described with a set of audio features having different time resolution still remains a challenging task. The feature matrices of different songs can be hardly involved in clustering algorithm directly. To tackle this issue, we model each feature dimension of one song following a so called "bag-of-features" approach [2]. Here feature values for each dimension are modeled by a single Gaussian, so that each feature dimension within a song is represented by the sample mean and standard deviation of the feature values. In addition, for each dimension of low-level and mid-level features we calculate the differences between the neighbor frames. This forms so called *delta* features that have already proved their efficiency for MFCCs. We likewise model each dimension of delta features with a single Gaussian. In addition, each feature dimension is normalized by mean and standard deviation. All in all, each music track is represented with a feature vector having 2342 feature dimensions.

Table 2: ISMIR 2004 benchmark dataset

Genre	Number of songs
Classical	320
Electronic	115
Jazz and Blues	26
Metal and Punk	45
Rock and Pop	101
World music	122

4.2 Dataset Description

In our experiments we use the "Training" part of the ISMIR2004 Audio Description Contest Dataset[1]. This dataset includes 729 music tracks that are manually subdivided into 6 genre categories as presented in Table 2.

In the context of this work genre labels are not directly employed in the traditional classification scenario. Instead of that we use the genre labels to generate constraints for the clustering algorithm. As such, two songs belonging to the same genre are considered to be connected with a *must-link* constraint. Likewise two songs that belong to different genres are connected with a *cannot-link* constraint. The details on the choice of the constrained songs are provided in Sec. 3.

4.3 Evaluation Scenarios

Traditionally constrained clustering is evaluated on the entire dataset – both on constrained and on non-constrained part – and the improvement of performance is shown over the number of pairwise constraints (see e.g. Basu et al. [3]). This approach is not optimal for the estimation of generalization capabilities of the clustering algorithm. Seeing the evaluation scores for the entire dataset, it is rather hard to estimate if the improvement is coming through the rising amount of constrained songs or through the general improvement of clustering quality. In addition to the common scores for the entire dataset (further denoted as *All* dataset) we perform the evaluation on the part of the dataset that is not involved in any constraints (further denoted as *Test* dataset).

Interpretation of the number of pairwise constraints is also not trivial. For instance, ten pairwise constraints can involve just five songs if the constraints are provided in a manner of a complete graph. On the other hand, ten pairwise constraints can also concern twenty songs if each constraint connects a distinct pair of songs. Instead of the number of the pairwise constraints we account for the percentage of the dataset involved in constraints.

4.4 Evaluation Measures

We have applied several metrics for cluster evaluation. One of the most traditional evaluation measures for clustering [3] is *normalized mutual information* (NMI). NMI is an information-theoretic measure which shows the amount

[1] `http://ismir2004.ismir.net/genre_contest/index.htm`

of information shared by ground-truth cluster assignments (represented with a random variable Y) and estimated cluster assignments (represented with a random variable Z):

$$NMI = \frac{2 \cdot I(Y;Z)}{H(Y) + H(Z)} \ , \tag{8}$$

where $I(Y;Z) = H(Y) - H(Y|Z)$ is the mutual information between Y and Z, $H(Y)$ is a marginal entropy of Y, and $H(Y|Z)$ is the conditional entropy of Y given Z.

As additional information-theoretic evaluation measures we use the normalized conditional entropies by Lukashevich [13] developed for evaluating song segmentation. These scores – in the context of this paper named *over-clustering* (S_o) and *under-clustering* (S_u) – give some insights to the origin of the clustering errors. The errors caused by the fragmentation of true clusters are captured by over-clustering S_o defined as

$$S_o = 1 - \frac{H(Z|Y)}{log_2 N_Z} \ , \tag{9}$$

and erroneous connection of elements of different clusters into one cluster is reflected by under-clustering S_u

$$S_u = 1 - \frac{H(Y|Z)}{log_2 N_Y} \ , \tag{10}$$

where N_Y and N_Z is a number of clusters in ground-truth and estimated cluster assignments respectively.

Pairwise F-measure is defined as the harmonic mean of pairwise precision and pairwise recall. Let M_Y be a set of song pairs that are in the same cluster in the ground-truth clustering, ex. pairs of songs having the same genre label. Likewise let M_Z be a set of identically labeled song pairs that are in the same cluster according to the estimated cluster assignments. Then pairwise precision (P_p), pairwise recall (R_p), and pairwise F-measure (F_p) are defined as

$$P_p = \frac{|M_Z \cap M_Y|}{|M_Z|} \ ,$$

$$R_p = \frac{|M_Z \cap M_Y|}{|M_Y|} \ ,$$

$$F_p = \frac{2 \cdot P_p \cdot R_p}{P_p + R_p} \ , \tag{11}$$

where $|\cdot|$ denotes the number of the corresponding pairs. Note that Basu et al. [3] used a slightly modified definition of pairwise F-measure, where they considered only the pairs of points that do not have explicit constraints between them. In our case we do not embed this information explicitly into pairwise F-measure. In contrast, we make difference between two evaluation scenarios – entire dataset and test part not involved in constraints – as described in Sec. 4.3.

To simplify the comparison with the work of Peng et al. [15] we additionally take into consideration *accuracy* and *purity* performance measures.

5. RESULTS

In this section we present the results for the experiments stated in Table 1 of Sec. 3. Each of the experiments is run over the following quantities of features: 16, 32, 64, 128, 256 and 512 determined through the powers of two. This log-line scale is used considering that improvement is more significant when we only have a small number of features. For a given number of features, the percentage of songs involved in constraints is augmented by five percent in each step, starting with 0 and stopping at 75.

Its worth to note, that all experiments with random selection of songs to be constrained, have been run 10 times, and that all clustering evaluation measures are the means of these runs. In addition, a random base line clustering was used as a reference, where the items of each ground-truth class were randomly uniformly distributed over k estimated clusters. Resulting values of normalized mutual information are presented in Fig. 3. It is possible to note that the RAW scores tend to be more smooth over the incrementing size of the constrained set than scores for the outlier identification methods. Clustering results of LOF and RRS for small number of constraints – constrained data set smaller than 30% – from both Symmetric and Random Walk Laplacians are comparable to the clustering results with random constraints. On the other hand, for the high number of constraints and the high amount of features, the clustering results of both LOF and RRS and for both Laplacians are significantly better than clustering results with random constraints, bringing an improvement of up to 0.22 points of NMI.

In fact, with more than 32 features the results are considerably better for almost all sets of constraints. Clustering with RRS seems to suffer from some instability, yet the differences between RRS an RAW with the Random Walk Laplacian are considerable while taking into account more than 32 features.

We present the scores of clustering evaluation measures for all experiments in Fig. 2. As a representative example we look at the clustering results with 512 feature dimensions. Here the scores for Symmetric Laplacian with RAW and with LOF are considerably lower. On the other hand, the best results are obtained from RRS with both Random Walk and Symmetric Laplacians.

6. CONCLUSIONS

In this paper we presented a system for the active exploration of music collections via spectral clustering with constraints. For the experiments we simulated the constraints through the ground-truth class labels of the audio genre dataset. Alongside with determining the constraint candidates in a random manner, we investigated two different outlier identification methods. Additionally we looked into a spectral feature selection method and proved the performance of clustering for two versions of Laplacian for spectral clustering.

7. ACKNOWLEDGMENT

This work has been partly supported by the German research project *GlobalMusic2One*[2] funded by the Federal Ministry of Education and Research (BMBF-FKZ: 01/S08039B). Additionally, the Thuringian Ministry of Economy, Employment and Technology supported this research by granting funds of the European Fund for Regional Development to the project *Songs2See*[3], enabling transnational cooperation between Thuringian companies and their partners from other European regions.

[2]see http://www.globalmusic2one.net
[3]see http://www.songs2see.net

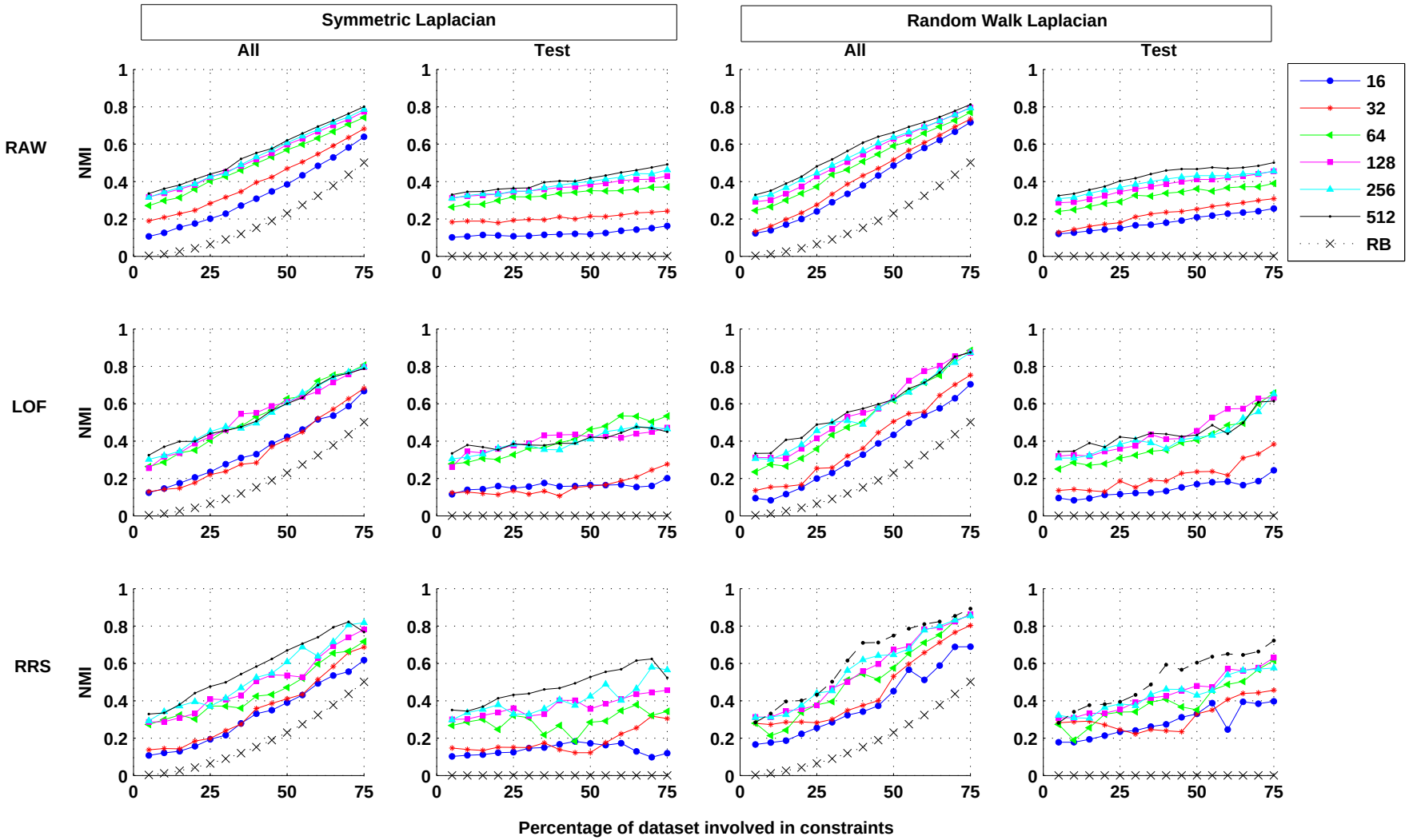


Figure 3: Normalized Mutual Information (Y axis) versus the percentage of the dataset that is involved in at least one constraint (X axis), different number of selected features and different evaluated subsets (*All* data set and *Test* set). Each row is related to a particular outlier identification method. The two first columns are related to the Symmetric Laplacian and the following two columns to the Random Walk Laplacian. Results of columns 1 and 3 are related to the *All* data set, while columns 2 and 4 are related only to the Test data set.

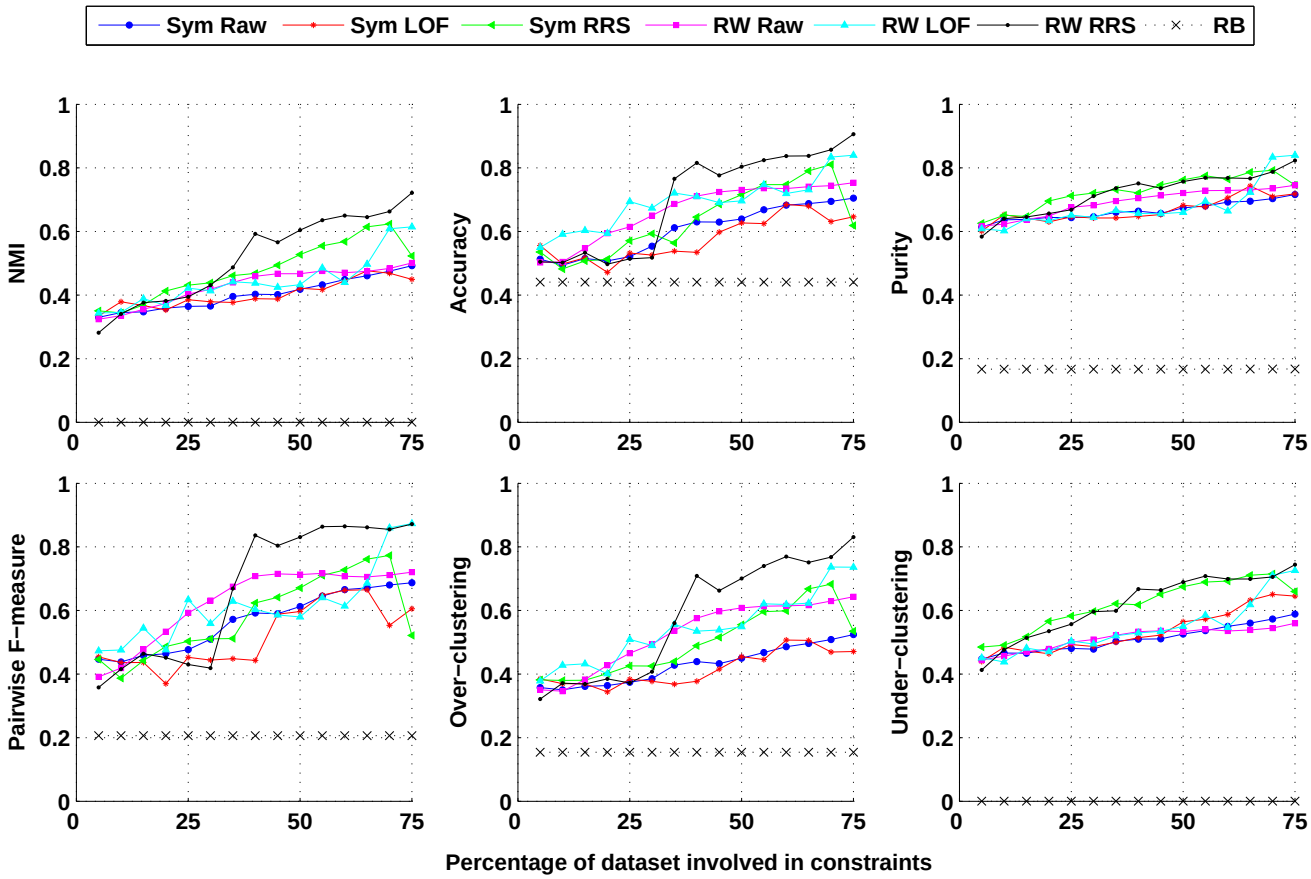


Figure 2: Values of several evaluation measures versus the percentage of the dataset that is involved in at least one constraint (X axis). Evaluation measures, starting from upper left plot and going to lower right plot: Normalized Mutual Information, Accuracy, Purity, Pairwise F-measure, Over-Clustering and Under-Clustering. In each of these plots experiments presented in Table 1 are evaluated. The number of selected features is fixed to 512. Curves plotted with 'crosses' state for random baseline clustering.

8. REFERENCES

[1] L. Atlas and S. S. Shamma. Joint acoustic and modulation frequency. *EURASIP Journal on Applied Signal Processing*, 2003:668–675, 2003.

[2] J.-J. Aucouturier, B. Defreville, and F. Pachet. The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. *J. Acoust. Soc. Am.*, 122(2):881–891, 2007.

[3] S. Basu, A. Banjeree, E. Mooney, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM-04)*, pages 333–344, 2004.

[4] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 59–68, 2004.

[5] J. P. Bello and J. Pickens. A robust mid-level representation for harmonic content in music signals. In *Proc. of the 6th Int. Conf. on Music Information Retrieval (ISMIR)*, 2005.

[6] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. *ACM SIGMOD Record*, 29(2):93–104, June 2000.

[7] C. Dittmar, C. Bastuck, and M. Gruhne. Novel mid-level audio features for music similarity. In *Proc. of the Int. Conf. on Music Communication Science (ICOMCS)*, Sydney, Australia, 2007.

[8] G. Gatzsche, M. Mehnert, D. Gatzsche, and K. Brandenburg. A symmetry based approach for musical tonality analysis. In *Proc. of the 8th Int. Conf. on Music Information Retrieval (ISMIR)*, Vienna, Austria, 2007.

[9] M. Gruhne and C. Dittmar. Comparison of harmonic mid-level representations for genre recognition. In *Proc. of the 3rd Workshop on Learning the Semantics of Audio Signals (LSAS)*, Graz, Austria, 2009.

[10] M. Gruhne, C. Dittmar, and D. Gaertner. Improving rhythmic similarity computation by beat histogram transformations. In *Proc. of the 10th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Kobe,

Japan, 2009.

[11] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: a kernel approach. In *Proc. of the 22nd Int. Conf. on Machine Learning (ICML)*, 2005.

[12] K. Lee. Automatic chord recognition from audio using enhanced pitch class profile. In *Proc. of the Int. Computer Music Conf. (ICMC)*, New Orleans, USA, 2006.

[13] H. Lukashevich. Towards quantitative measures of evaluating song segmentation. In *Proc. of the 9th Int. Conf. on Music Information Retrieval (ISMIR)*, pages 375–380, Philadelphia, USA, 2008.

[14] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

[15] W. Peng, T. Li, and M. Ogihara. Music clustering with constraints. In *Proc. of the 8th Int. Conf. on Music Information Retrieval (ISMIR)*, Vienna, Austria, 2007.

[16] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 427–438, 2000.

[17] M. Stein, B. M. Schubert, M. Gruhne, G. Gatzsche, and M. Mehnert. Evaluation and comparison of audio chroma feature extraction methods. In *Proc. of the 126th AES Convention*, Munich, Germany, 2009.

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

[19] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[20] C. Uhle, C. Dittmar, and T. Sporer. Extraction of drum tracks from polyphonic music using independent subspace analysis. In *Proc. of the 4th Int. Symposium on Independent Component Analysis (ICA)*, Nara, Japan, 2003.

[21] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *Proc. of the 17th Int. Conf. on Machine Learning (ICML)*, pages 1103–1110, 2000.

[22] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *Proc. of the 24th Int. Conf. on Machine Learning (ICML)*, 2007.

Music Recommendation in the Personal Long Tail:
Using a Social-based Analysis of a User's Long-Tailed Listening Behavior

Kibeom Lee
Graduate School of Culture Technology, KAIST
Daejeon, Korea
kiblee@kaist.ac.kr

Woon Seung Yeo
Graduate School of Culture Technology, KAIST
Daejeon, Korea
woon@kaist.ac.kr

Kyogu Lee
Department of Digital Contents Convergence, Seoul National University
Seoul, Korea
kglee@snu.ac.kr

ABSTRACT

The online music industry has been growing at a fast pace, especially during the recent years. Even music sales have moved from physical sales to digital sales, paving the way for millions of digital music becoming available for all users. However, this produces information overload, where there are so many items available due to, virtually, no storage limitations, it becomes difficult for users to find what they are looking for. There have been many approaches in recommending music to users to tackle information overload. One successful approach is collaborative filtering, which is currently widely used in commercial services. Although collaborative filtering produces very satisfying results, it becomes prone to popularity bias, recommending items that are correct recommendations but quite "obvious". In this paper, a new recommendation algorithm is proposed that is based on collaborative filtering and focuses on producing novel recommendations. The algorithm produces novel, yet relevant, recommendations to users based on analyzing the users' and the entire population's listening behaviors. An online user test shows that the system is able to produce relevant and novel recommendations and has greater potential with some minor adjustments in parameters.

Categories and Subject Descriptors

I.1.2 [**Computing Methodologies**]: Algorithms – *Nonalgebraic algorithms, analysis of algorithms*

General Terms

Algorithms

Keywords

Recommender systems, collaborative filtering, music recommendation

1. INTRODUCTION

With advances in the Internet, lower hardware costs, increasing peer-to-peer networks, and the popularity of high-storage portable media players, the online music industry has been growing rapidly, especially during the past few years. Gradually, music sales have moved from physical album sales to digital sales from online stores. Currently, these services offer millions of tracks to users, the catalog growing rapidly in size compared to the size when the services were first announced. For instance, Amazon offered over 2 million songs to users when the music service launched, but now offers over 11.8 million songs as of 2010. Some notable online music stores, including Amazon, are Amazon MP3 (11,000,000+ songs), iTunes Store (12,000,000+ songs) and Rhapsody (9,000,000+ songs). Apart from music stores, there are also music streaming services that offer millions of songs, such as Lala (8,000,000 songs), Spotify (8,000,000 songs), and Last.fm (7,000,000 songs).

These large numbers of songs available to users are a result of the Long Tail business model [1], contrary to only products that were in demand being sold in stores. However, as a result, although paradoxical, users have ended up listening to less music now that so much is available, simply because it is hard to find new and relevant music. For instance, digital track sales surpassed the 1 billion sales mark in 2008. However, the Top 200 digital tracks alone accounted for 17% of the entire track sales (184 million sales) [2].

WOMRAD 2010 Workshop on Music Recommendation and Discovery, colocated with ACM RecSys 2010 (Barcelona, SPAIN)

2. RELATED WORK

2.1 Collaborative Filtering-based Recommender Systems

One of the earliest recommender systems based on collaborative filtering is Tapestry [3]. Stemming from the need to handle increasing numbers of emails, Tapestry used explicit opinions of people in a relatively small group, such as an office workgroup, to filter out incoming email for a given user. However, a drawback of this system was that users had to be familiar with the preferences and opinions of other people in their network, which is why Tapestry worked on small networks like the office.

A more general collaborative filtering approach was developed by Resnick et al. called GroupLens [4]. The basic idea behind GroupLens, which aimed to help users find news articles amongst the vast available numbers, was that "people who agreed in the past will probably agree again". Using this heuristic, the GroupLens system was able to predict the ratings of certain news articles by a given user. An advantage that this provided was that the collaborative filtering could be scaled, unlike Tapestry, because a user was not required to actually know other users that had similar preferences to him. This was done by the system, which gathered information on the ratings of users, naturally

creating another advantage of users being anonymous inside the whole system.

Research related to, and including, the above studies focused on filtering a vast amount of text, which were in forms of emails, news, and messages, to those that were worth reading. Items would be given to the user with their prediction scores, aiding the user in which item to read next. The next wave of studies focused on a more direct approach in recommending items.

Ringo was a system developed to provide personalized music recommendations [5]. It maintained a user's profile, a history of ratings on various artists that were essentially explicit labelings on which artists the user does or does not enjoy listening to. These profiles were matched by the system to calculate recommendations on which artists had the highest probabilities of being liked by the user.

While Ringo was focused on music items, Bellcore's recommender system focused on movies [6]. Like Ringo, it used a database of movie ratings by users and matched rating profiles to provide recommendations by finding similar users and the movies that they had watched and rated positively. Tests on the reliability of the recommender system showed that three out of every four recommendations would be rated highly by the user, and also showed that the system produced extremely more accurate recommendations compared to nationally-known movie critics.

While there were numerous advances and algorithms related to collaborative filtering since then, the most well-known collaborative filtering system today, however, is probably the system used in Amazon.com, an electronic commerce company that sells books, movies, music, etc. Amazon.com offers recommendations on items that are similar to the item being purchased, rather than finding similar users and then recommending the items those users have purchased. This method, which is called item-to-item collaborative filtering, scales to extremely large datasets and generates satisfiable results.

2.2 Collaborative Filtering-based Recommender Systems for Music

Although the collaborative filtering-based approaches above were designed on specific items, the algorithms can be generalized and applied to music recommendation. Hence, the results of such algorithms applied to music are not much different than applied to the original items.

Apart from recommender systems that use data on the ratings and/or purchases of items, there are other collaborative filtering-based recommender systems that take advantage of metadata produced by users that are found in music.

[7] presents some examples of metadata used in such algorithms, which include reviews, lyrics, blogs, social tags, bios, and playlists. Examples of commercial services that use such approaches are Rate Your Music (reviews), The Hype Machine (blogs), last.fm (social tags), and playlist.com (playlists).

Social tags, a representative product of online collaboration, has been used heavily in music recommendation systems. Hu and Downie explored the mood metadata associated with songs and their relationships with music genre, artist, and usage metadata [8]. They found that the genre-mood relationships and artist-mood relationships showed consistencies, showing the potential of being utilized in automated mood classification tasks. Eck et. al proposed a method for generating social tags for music that lack such tags [9]. Audio features of songs were analyzed and mapped to tags, using a set of boosted classifiers. These were then utilized on untagged songs, populating them with the associated social tags depending on the musical content. This enables unpopular songs and/or new songs that have no social tags to be used in music recommenders that use a social algorithm. It also tackles the cold start problem, a problem found in collaborative filtering-based recommender systems. Symeonidis et. al analyzed social tags in order to tackle the problem of the multimodal use of music [10]. They developed a framework that modeled users, tags, and items, altogether. This was then used in recommending musical items (artists, songs, and albums) to users by performing latent semantic analysis and dimensionality reduction according to each user's multimodal perception of music. Levy and Sandler inspect the seemingly ad hoc and informal language of tagging as a high-volume source of semantic metadata for music. Results show that tags establish a low-dimensional semantic space, being extremely polished at the track level, especially by artist and genre. Using these results, the authors also introduce an interface for users to browse by mood, through a two-dimensional subspace that represents musical emotion.

Celma introduces a system that recommends music and the relevant information associated with the recommended music [11]. The proposed system uses the *Friend of a Friend* and RSS vocabularies for creating recommendations, taking in consideration the user's musical tastes and listening habits. The FOAF project provides protocols and a language to describe homepage-like content and social networks, ultimately providing the proposed system with the user's profile. The RSS vocabulary provides the system with syndicated content, which includes data such as new album releases, album reviews, podcast sessions, upcoming gigs, etc. Thus, the proposed system improves the existing recommendation systems by understanding the users through psychological factors (personality, demographic preferences, socioeconomics, situation, social relationships) and explicit music preferences.

3. LIMITATIONS OF COLLABORATIVE FILTERING

3.1 Popularity Bias

Collaborative filtering-based recommender systems produce good results and are used widely in commercial services such as Amazon.com and Last.fm. However, collaborative filtering has some common limitations that occur naturally due to its roots lying in the wisdom of crowds. One of the largest problems of collaborative filtering is popularity bias [12, 13].

This happens when a popular item is associated with many other related items. Users that interact with these items are then recommended the popular item. The system recommends the popular item often, leading to item purchases (or any other form of positive input from user) and as this item is purchased more, it is also recommended more. This loop, in which the rich become richer, creates popularity bias.

Naturally, as a result of the above feedback loop, the recommender system tends to bias its recommendations towards popular items. Thus, the recommendations lose their novelty [12, 13] and make it extremely difficult to recommend lesser-known artists.

In Amazon.com, in which collaborative filtering is heavily used, the popularity bias can be seen when viewing the recommendations that are offered when searching for popular items. For instance, the 98 recommendations that appear when searching for Harry Potter includes The Da Vinci Code, To Kill a Mockingbird and 28 other Harry Potter books and DVDs. In the case of music, searching for The Beatles' Revolver album results in 33 albums from The Beatles, out of a total of 97 recommendations, as shown in Figure 1. The other recommended items show well-known artists that user's, who are interested in The Beatles, will most likely have heard of already such as The Rolling Stones, Led Zeppelin, and Neil Young. These recommended artists are *correct* recommendations but fail to be novel recommendations.

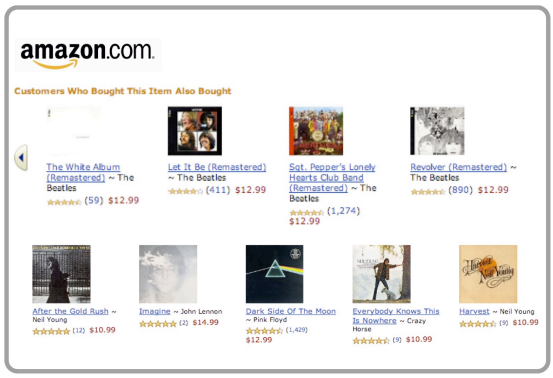


Figure 1. Recommendations from Amazon.com, which are all quite "obvious" recommendations, although they are correct recommendations.

Due to this popularity bias, a large portion of the recommended items result in obvious recommendations that may be relevant to easy-going, casual listeners, but not so helpful for enthusiastic music listeners, who have a high probability of already being knowledgeable on the artists being recommended.

The number of high quality, or "correct", recommended items that are produced with collaborative filtering is verified by [14]. However, the problem of popularity bias was also verified as the amount of novel recommendations given to a user was the lowest for collaborative filtering in an experiment comparing collaborative filtering, content-based, and hybrid methods [14]. Thus, it was confirmed that collaborative filtering results in less percentage of novel songs but of higher quality.

4. ALGORITHM

In this section, we provide an algorithm that is based on collaborative filtering, yet overcomes popularity bias, a natural problem that arises from CF. Also, the algorithm focuses on providing recommendations that are novel to the user, while also remaining relevant.

To implement this algorithm, user data from Last.fm, an Internet service that provides users with streaming music via radio stations, was used. Reasons for selecting Last.fm was the readily available developer API and the various, massive amount of data that was available such as user playlists, playcounts for artists and individual songs, artist information, song information, and most importantly, the worldwide popularity of the site.

4.1 Concept of Recommendation Algorithm

4.1.1 Changing Perspectives on Novel Recommendations

While the goal of recommenders in general is to provide recommendations that are novel and relevant to the user, as stated beforehand social-based recommendations, although relevant, fail in providing novel recommendations to users. In contrast, content-based recommender systems work better in providing novel recommendations because they are not affected by popularity or any other social influence [15].

Another method to provide novel recommendations to users is to use the long tail popularity distribution of the artists [7]. This idea can be applied to both content-based and social-based algorithms. Content-based algorithms can use the long tail distribution to recommend similar items based on content-analysis and also found in the tail portion of the distribution. For social-based algorithms, or collaborative filtering, the idea can be applied by first obtaining the full list of recommendations and then removing the recommendations that lie in the head portion of the distribution. This would result in recommendations being novel to the user, since it is unlikely that artists residing in the tail portion of the distribution are known to the user.

However, although strictly recommending artists from the long tail and avoiding recommending those that are obvious (those that are located in the head portion of the distribution) have a high probability of being novel recommendations, we need to take in consideration that novel recommendations are relative to the user. In other words, it is naive to assume that the user will be aware of certain artists just because they are in the head portion of the long tail distribution. Thus, the fact that even popular artists have a possibility of being novel recommendations to certain users must not be overlooked.

4.1.2 User Listening Behavior

As shown in Figure 2, which shows a random Last.fm user's playlist in descending order of playcount, the listening behavior shows a distribution that is similar to that of long-tail distributions. Users tend to listen to an extremely small portion of their playlists often while the remaining songs seldom get played. Due to the data available, which is the top 500 played songs in the user's playlist, all of the songs in the graph are played at least once.

Figure 2. The listening behavior of a user and his/her entire playlist. Although not exact, the graph shows a long-tailed distribution where the majority of tracks are seldom played.

4.1.3 Defining Experts and Novices

Using this long-tailed distribution of users' listening behaviors, the users can be divided into two groups: experts and novices. Here, users are considered "experts" regarding the songs/artists that they listen to often, i.e. songs/artists that lie in the head portion of the long-tailed listening behavior. On the other hand, users are considered "novices" regarding the songs that reside in the tail portion.

4.1.4 The Mystery of Unpopular "Loved" Songs

Last.fm provides users with an option to mark songs "loved" (Figure 3). This kind of feedback from users explicitly shows that a user enjoys a particular song. One would expect that these "loved" songs would all lie in the head portion of the listening behavior distribution. However, these songs that are marked "loved" can be found scattered throughout the entire distribution. Here, a paradox can be found: Why are some songs marked "loved" lying at the tail end of the playcount distribution? One would assume that a "loved" song would have a high playcount, but a quick inspection shows that this is not the case. Thus, an assumption that is made here, a key assumption in this algorithm, is that songs are marked "loved", yet remain in the tail, because the user is unfamiliar with that song/artist/genre, i.e. is a novice, but happened to stumble upon that particular song and liked it.

Figure 3. The "tail" portion of a random user's playlist. There are two songs marked "loved" by the user, but have only been played three times.

Among the 21,688 users whose data was used for the algorithm, 78.3%, or 16,973 users, used the "love" function provided in Last.fm. Among the 16,973 users who utilized the "love" function, 77.8% of the users had "loved" songs in the tail portion of their playlist's song distribution sorted by playcount.

Upon closer inspection of the random user in Figure 3, the songs/artists in the "head" portion came from various genres such as electronic, hip-hop, and reggae. What they did have in common, however, was that they were all German artists, including the user herself. Looking at the songs that were marked "loved" but were not played often, we can see that they too come from different genres, but are both artists from the U.S.

The previously mentioned assumption that fuels this algorithm was made after observing such occurrences in users' playlists. According to our assumption, we assume that the user, who is German, is a novice in artists from the U.S. and stumbled across several songs that she liked. However, she did not get to venture similar songs and/or artists because she was unaware of which artists/songs were similar.

4.1.5 The Big Picture

Once the basic assumptions are made and the new definition of novices and experts are established, the concept of the recommendation algorithm can be explained. As shown in Figure 4, recommendations can be made to novices of certain song sets using the information that can be obtained by a group of experts that have those song sets in the head portion of their listening behavior distribution.

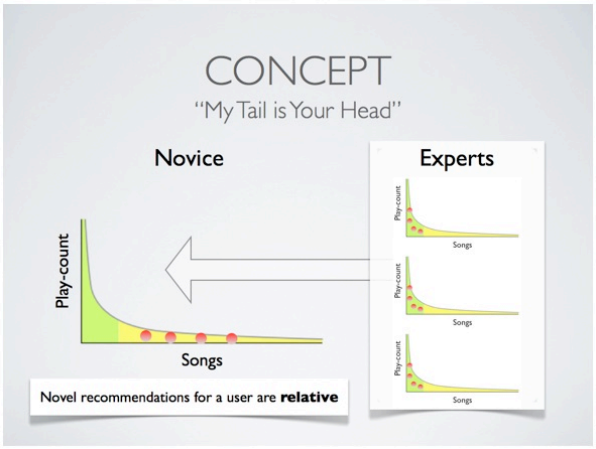


Figure 4. The overview of the algorithm showing the concept of novices and experts.

By using the listening behavior of experts to provide recommendations to novices, the recommended items will be novel to the user, contrasting to other recommendation systems that simply recommended artists/songs from the tail of the popularity distribution of items. In other words, while remaining novel to the specific user, the recommended items may or may not be in the far, unpopular end of the popularity distribution. In fact, even popular items that reside in the head of the popularity distribution may be recommended, but the user may not be aware of the recommended item since the recommendations were based on the user's tail portion of her listening behavior distribution, in which the user was considered a novice.

In addition to being novel recommendations, the recommended items will also be relevant to the user since the recommendations were found using songs that the user had marked "loved", explicitly stating the user's view on that particular item, and then using collaborative filtering to find experts on those "loved" songs to find relevant recommendations.

4.2 Data

User data was collected in order to test the algorithm and evaluate the results of the recommendations from early March to late April in 2010. Data was collected from the Last.fm website using a custom web crawler and the Last.fm API. The user data that was collected included the songs that the user had listened to overall, meaning the songs that the user listened to from the day he/she registered at Last.fm up until the day the data was collected. It also included the playcount for each song, song title, artist name, user ID, rank, and whether it was marked "loved" or not. The data that was collected is summarized below in Table 1.

Table 1. Summary of amount of data collected

Data	Count
Users	21,681
Unique Songs	2,001,324
Songs from All Playlists	9,073,681

4.2.1 Last.fm API

All the collected information, except the playlist history, was gathered via the Last.fm API. Although the algorithm could have queried the information in real-time, it was decided that having local data would facilitate in quicker results. After fetching the data, we had song titles and corresponding artist names of approximately 2 million songs.

In addition to the user and song data collected with the Last.fm API, artist popularity was also measured indirectly via the API. Because the Last.fm API did not provide the artist ranking directly through the API, we had to collect the number of Listeners and Plays, which were offered through the API. By having the Listeners and Plays of a given artist, we would be able to determine the overall ranking of popularity of the artists. This will be further explained in the next section.

4.2.2 User Data Crawler

Unfortunately, the Last.fm API query for a given user's listening history returns the top 50 songs ordered by playcount. This was not adequate enough since the algorithm needed the entire playlist in order to utilize the long tail of the playcount distribution.

In order to solve this problem, a custom crawler was implemented to collect the users' listening history (referred to as 'playlist' in this paper) and playcount information. Although this returned a maximum of 500 results (Last.fm displays only top 500 songs in the playlist), the data was adequate to be divided into the short head and long tail and used in the algorithm.

Data on a total of 21,681 random users were crawled. The playlists and the according information were also stored for each user, resulting in 21,681 playlists with a total of 9,073,681 songs. Because playlists from different users contain lots of duplicate entries, the number of unique songs that were crawled, as stated above, was 2,001,324 unique songs.

4.3 Algorithm

As shown in Listing 1, the user that will receive the recommendations, whom we will call "novice" according to the algorithm's concept, is given as input to the algorithm. Then, the listening behavior for the novice is retrieved using data available at Last.fm. As long as the user is not a new user and has been listening to his/her playlist, the playcount distribution of his/her playlist is more than likely to show a long-tailed distribution, in which a small set of songs have been listened with a heavily biased frequency while the remaining songs listened only occasionally. Since we are interested in the songs/artists that the given user is a novice on (i.e. songs marked "loved" in the long tail), we discard the head portion of the distribution and from the remaining songs, which are songs in the tail portion, we discard all songs except those that are explicitly labeled "loved" by the novice. These remaining songs, denoted by 'S_1', will be the song set that will be used to create recommendations.

Next, using the listening behavior of the other users from our database, we find those that listen to the songs in song set S. In other words, we find the "experts" on song set S by finding users that have a subset of song set S in the head portion of their listening behavior distribution. If such users exist, we compare the songs in the "head" of their playcount distribution with song set S and use the remaining, non-overlapping songs as recommendation candidates and assign the weight for those items according to the strength of the match between the songs in the expert's "head" and song set S.

```
begin Recommendations REC (aGivenUser U1);
  do
    Result R1 := retrieveListeningBehaviorDistribution(U1);
    SongSet S1 := getSongsInLongTail(R1);
    S1_loved := filterLovedSongs(S1);
    for i := 2 to n (n: number of users) step 1 do
      Result Ri := retrieveListeningBehaviorDistribution(Ui)
      SongSet Si := getSongsInHead(Ri);
      if (Si ∩ S1 ≠ ∅) do
        CandidateSongSet CSi := (Si ∪ S1) – (Si ∩ S1);
        incrementWeight(CSi);
        REC += CSi;   od
      od
    od
    printRecommendations();
```

Listing 1. Pseudoalgorithm for proposed recommender system.

These recommendation candidates are accumulated in the global song set REC, and the weight of the candidate are incremented as they are recommended to REC. Finally, the recommendations are given to the user in the order of their weights.

4.4 Parameters

The algorithm is quite flexible as it has many parameters that can be changed, which greatly influences the recommended items to the user. Parameters that play a crucial role in the overall quality of the recommendations include:

- The size of the "head" of experts
- The size of the "tail" of novices
- Weights of recommended items

4.4.1 Expert Parameter

The parameter that influences the outcome most is the size of the "head" portion of the expert's listening behavior distribution. For example, if the value for this parameter is set to "10", a user is considered an expert only if the top ten songs that s/he listened to contains any number of songs from the set of songs that are marked "loved" in the novice's "tail" portion of his/her listening distribution. In other words, this parameter determines the qualification strictness on which users are considered experts.

The lower the value, the harder it is for a given user to be considered an expert. Also, as the value is lower, the resulting recommendations are more novel, in contrast to when the values are higher, in which the resulting recommendations become those that are well-known. As the value is set higher, the recommendations represent those that are from the existing music recommendations that are offered using traditional collaborative-filtering methods.

4.4.2 Novice Parameter

The parameter that can be varied for the novice users is the size of the "tail" portion of the novice's listening behavior distribution.

Opposite of the expert parameter, the novice parameter sets the range of songs in the user's playlist that the user is a novice on. Using loved songs that lay near the "head" portion may result in songs that the user is aware of, leading to the recommendations being less novel to the novice. However, this parameter does not have as much influence as the expert parameter has because once the novice parameter is set, the entire range of songs are not used, but only those that are explicitly marked "loved" by the user.

4.4.3 Weights of Recommended Items

A formal set of rules and equations to assign weights to the recommended items can greatly change the songs that will be presented to the user as recommendations. This is important because it is inappropriate to present the entire collection of songs that result from the algorithm, as the number may vary depending on the two parameters above. Among the final song set that contains hundreds of candidate songs for recommendations, only a subset, namely the top N songs are presented to the user. Thus, assigning the appropriate weights for these candidates can ultimately influence the outcome of the recommended items. Currently, the algorithm uses a simple approach in which the weight is equal to the number of times a song is a member of both the head of an expert and tail of the novice.

5. USER TEST & EVALUATION

There are many ways to evaluate a recommender system, both offline and online. A common online method to evaluate a recommender system is to generate test sets to be evaluated later [16]. Another popular method is to use cross-validation, in which the data is partitioned and used as test sets [17].

5.1 Difficulties in Evaluating Novel Recommendations

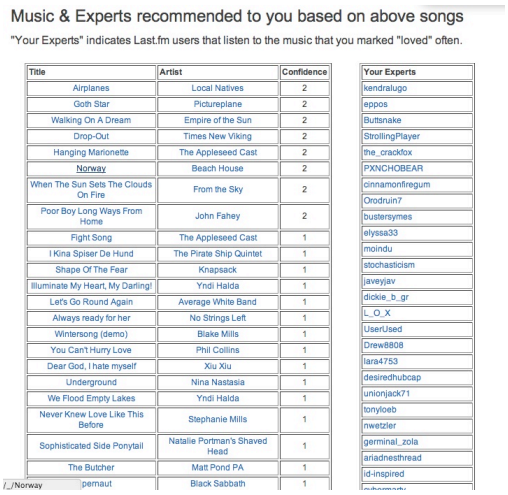
However, offline evaluations are not appropriate for recommender systems where the recommendations of novel items are important. This is because when a truly novel item is actually recommended to a user, meaning that the user does not already know about this item, it is extremely difficult for the user to evaluate the unknown item without providing any additional information [18]. Because of this, measuring novelty in the recommended items is a rather challenging task, leaving no option but to carry out live user studies where the users explicitly indicate whether the provided recommendations were novel or not [19].

Thus, in order to measure the novelty and relevance of the recommended items, an online user test was carried out using a fully functional website, including a section for explicit user feedback regarding the recommendations given to the users.

5.2 Design

A fully functional website was created in order to perform an online evaluation of the recommendations for random users. On the website, a user has to sign-up and input his/her Last.fm ID. After receiving a new ID, the server runs the recommendation algorithm on that particular Last.fm ID. Meanwhile, the user was requested to come back shortly afterwards, while the recommendations were being processed. The algorithm had to be run in real-time online because of the nature of it being heavily dependent on the user information. Also, pre-calculating the recommendations for users in the local database offline and then providing them online was unrealistic as the probability that a new user would also be one that was pre-calculated was extremely low. When the user returns, he/she is presented with two sets of recommendations.

Recommendation Set 1 was the results of the algorithm with the Expert Parameter, the parameter that determines the size of the "head" portion of the expert, set to 5. A value of 5 for the Expert Parameter means that the algorithm is being very strict about which users are qualified to be experts. This produces dense novel items. Recommendation Set 2 was the results with the Expert Parameter set to 10. A value of 10 tends to mix novel recommendations and well-known recommendations, so is more of a general setting that aims to resemble recommendations from Last.fm. After the user views the recommendations, a survey page was available to provide explicit feedback on the quality of the recommendations given to them.


Music & Experts recommended to you based on above songs

"Your Experts" indicates Last.fm users that listen to the music that you marked "loved" often.

Title	Artist	Confidence
Airplanes	Local Natives	2
Goth Star	Pictureplane	2
Walking On A Dream	Empire of the Sun	2
Drop-Out	Times New Viking	2
Hanging Marionette	The Appleseed Cast	2
Norway	Beach House	2
When The Sun Sets The Clouds On Fire	From the Sky	2
Poor Boy Long Ways From Home	John Fahey	2
Fight Song	The Appleseed Cast	1
I Kina Spiser De Hund	The Pirate Ship Quintet	1
Shape Of The Fear	Knapsack	1
Illuminate My Heart, My Darling!	Yndi Halda	1
Let's Go Round Again	Average White Band	1
Always ready for her	No Strings Left	1
Wintersong (demo)	Blake Mills	1
You Can't Hurry Love	Phil Collins	1
Dear God, I hate myself	Xiu Xiu	1
Underground	Nina Nastasia	1
We Flood Empty Lakes	Yndi Halda	1
Never Knew Love Like This Before	Stephanie Mills	1
Sophisticated Side Ponytail	Natalie Portman's Shaved Head	1
The Butcher	Matt Pond PA	1
/_/Norway	pernaut	Black Sabbath	1

Your Experts
kendralugo
eppos
Buttsnake
StrollingPlayer
the_crackfox
PXNCHOBEAR
cinnamonfiregum
Orodruin7
bustersymes
elyssa33
moindu
stochasticism
javeyjav
dickie_b_gr
L_O_X
UserUsed
Drew8808
lara4753
desiredhubcap
unionjack71
tonyloeb
nwetzler
germinal_zola
ariadnesthread
id-inspired

Figure 5. Screenshot of the recommended items at the user-test website. Each facet of the recommended items are linked to pages at Last.fm for supplementary information

Since the goal of the algorithm is to provide novel recommendations, there had to be an easy way for the user to evaluate the recommended items, since it is assumed that if the recommended items are indeed novel, then the user has no knowledge about the item. Thus, each recommended item was hyperlinked to the according page in Last.fm, as shown in Figure 5. Through these links, users were able to evaluate the recommended items that were novel to them by visiting the linked pages. Last.fm provides related information regarding specific songs, which include music videos, song previews, and even a radio for the song's artist. By utilizing these pages, users were able to listen to the songs that were recommended to them.

5.3 Survey

On the survey page, a set of five questions were given to the user, each regarding one of the two sets of recommendation results that were produced by the algorithm. The questions were answered on a five-point Likert item. The final question was a subjective question, asking for any comments or feedbacks on the recommendations. The questions used in the survey are shown in Table 2.

Table 2. Questions used in the user survey.

Q. 1	How would you rate the relevance of items?
Q. 2	How would you rate the novelty of the recommended items?
Q. 3	How would you rate the serendipity of the recommended items?
Q. 4	How would you rate the recommendations overall?
Q. 5	Provide any comments/feedback about the recommendations that were given to you.

6. RESULTS & DISCUSSION

A user survey was carried out online accompanying the online music recommendation service because of the difficulties in measuring novelty. A total of 24 users tested the recommendations offered to them on the website. These users were random Last.fm users that had received private messages (advertising the user test) through the Last.fm messaging system. The new recommendation system was also advertised on various Last.fm groups whose interests were in finding new music or those who were unsatisfied with current recommender systems and their quite obvious recommendations. However, because the users had to answer two surveys for two different sets, some appeared to have quit abruptly after finishing the first set. As a result, only 11 users out of 24 completed the second survey.

The private messages were sent to random Last.fm users who satisfied two conditions: 1) the user used the "loved" function with his/her playlist, 2) The last time the user logged in was not more than two weeks ago from the day the private messages were sent. Despite the advertisements and private messages, the response rate was extremely low (< 10 %). The results are shown in Figures 6-8.

Figure 6. Comparison of the relevance ratings for the two sets

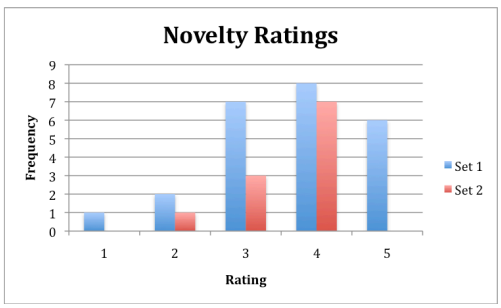


Figure 7. Comparison of the novelty ratings for the two sets

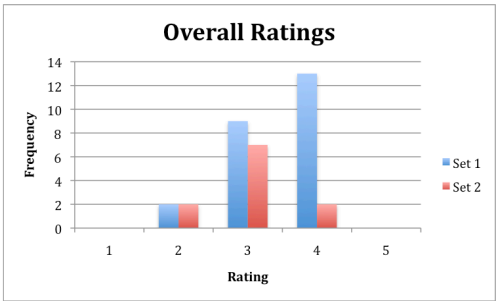


Figure 8. Comparison of the overall ratings for the two sets.

The results of the user test on the recommendations produced by the proposed algorithm are generally positive. The mean value for the relevance of the items was 3.417 (on a 5 point scale) with a confidence interval of 0.390 (with alpha value of 0.05). The mean values of novelty and serendipity were also on the positive side with 3.667 and 3.625, respectively. The confidence intervals were 0.436 (alpha = 0.05) for novelty and 0.350 (alpha = 0.05) for serendipity. The overall rating of the recommender system had a mean value of 3.458 with a confidence interval of 0.263 (alpha = 0.05). In general, the results show that the proposed system has positive ratings and could be refined to produce better results.

The proposed system was rated higher in both novelty and serendipity, compared to the second set of recommendations, which was a set of recommendations that was intended to imitate existing systems such as Last.fm.

For this study, the parameters of the system were set with values that we thought produced the desired results after several iterations of the algorithm. However, a full study focused on finding the optimal values for the parameters would be an excellent follow-up study and would greatly enhance the recommendations of the system.

The score for the novelty of recommended items could have been higher, because the algorithm did not check whether the recommended songs existed in the user's library before being offered. Thus, the user would see some artists that they were aware of. As implied above, it is quite easy to increase the percentage of novel items in the entire recommendation list: simply check whether the artist exists in the user's library and if it does, exclude it from the recommendations. However, this step was excluded from the algorithm deliberately to increase the confidence of the users on the proposed system. The basis for this was [20], in which the authors found that users liked to see familiar items in the recommendations, which ultimately led to an increase of user confidence in the system. Checking to see if the user is familiar with the recommended item would produce more "dense" novel recommendations.

Regarding the novelty of items, an unforeseen problem was revealed after the user test. One user commented, "I have most of the bands recommended on my computer, I just haven't given them much of a listen to. Grizzly Bear in particular..." The problem here is whether, in this user's case, Grizzly Bear is a novel recommendation. The user states that s/he did not listen to many of the recommended artists, although those artists were in his/her library. Because the algorithm depends on the playcount of the songs in a user's library it is totally blind to tracks that reside in the library but have a playcount of 0. Thus, it recommends songs that it believes to be novel to the user, when it could in fact

exist in the library already. Unsurprisingly, the novelty and serendipity ratings from this user were low (a score of 2 for each), but the rating on the overall system was positive (a score of 4). Clarifying such issues on what a novel item is would help improve the algorithm and the user's perception of the system.

7. FUTURE RESEARCH

The most urgent and important future work on this particular study would be to find the ideal parameter settings to produce the desired recommendations. Due to the available time frame for this study, much of the algorithm analysis including the settings of the parameters, were done manually, simply by iterating through different settings and observing the results. By finding the optimized values on parameters such as Expert Head Size, User Tail Size, and Item Weights, the quality of the recommendations in novelty and relevance would be greatly enhanced.

Work on expanding the flexibility of the algorithm can also be done, creating additional parameters that bring changes to the recommendations. More parameters would mean that the algorithm could be suited for each user's needs, bringing the possibility of creating an evermore-personalized set of recommendations.

The overall system itself could be further developed to integrate content-based analysis for better results. Although the proposed method is at its infancy, we believe that the only way to improve it further (after it has fully developed independently) will be to incorporate a content-based algorithm to improve on its remaining weaknesses as an algorithm that is based on user profiles.

8. CONCLUSION

In this paper, a novel approach to recommending unfamiliar artists relative to each user was proposed in order to tackle the problem of the high density of obvious items in the list of recommendations found in today's recommender systems. The key concept in this approach was that novel items did not always have to be items that reside in the long tail of the popularity distribution. Although novel or unfamiliar items, more often than not, do indeed reside in the long tail of the popularity distribution, it is important to acknowledge that even well-known artists could be unknown to users who are (a) interested in different genres and (b) are in different cultures and/or countries.

A system that produced recommendations was implemented and was available online for users to use and rate. The recommendations were produced using data collected from Last.fm. Results of the user surveys show that the proposed system succeeds in providing novel recommendations to users, while keeping those items also relevant. This study shows the potential of such an approach to recommending novel items, while maintaining a collaborative filtering algorithm without the support from content-based algorithms.

9. ACKNOWLEDGEMENTS

The authors would like to thank Professor Sangki 'Steve' Han at the Graduate School of Culture Technology, KAIST and Sheayun Lee for their valuable comments and feedback.

10. REFERENCES

[1] Chris Anderson. The Long Tail: Why the Future of Business Is Selling Less of More. Hyperion, 2006. ISBN 1401302378.

[2] Nielsen Soundscan, State of the Industry. *National Association of Recording Merchandisers*, 2008

[3] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. Using Collaborative Filtering to Weave an Information Tapestry. *Commun. ACM,* 35(12):61-70, 1992. ISSN 0001-0782.

[4] Resnick P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. Grouplens: An Open Architecture for Collaborative Filtering of Netnews. In *CSCW 1994,* pages 175-186.

[5] Shardanand, U. and Maes. P. Social Information Filtering: Algorithms for Automating "word of mouth". In *CHI `95*, pages 210-217.

[6] Hill, W., Stead, L., Rosenstein, M., and Furnas, G. Recommending and Evaluating Choices in Virtual Community of Use. In *CHI `95,* pages 194-201.

[7] Celma, O. and Lamere, P. If you like the Beatles you might like…: a tutorial on music recommendation. *ACM Multimedia*, pages 1157-1158, ACM, 2008.

[8] Hu, X and Downie, J. S. Exploring mood metadata: Relationship with genre, artist, and usage metadata. *,* September 2007.

[9] Eck, D., Lamere, P., Bertin-Mahieux, T., and Green, S. Automatic generation of social tags for music recommendation. In *Advances in Neural Information Processing Systems 20.* MIT Press, 2008.

[10] Symeonidis, P., Ruxanda, M. M., Nanopoulos, A., and Manolopoulos, Y. Ternary semantic analysis of social tags for personalized music recommendation. *ISMIR*, pages 219.

[11] Celma, O. Foafing the music: Bridging the semantic gap in music recommendation. In *Proceedings of the 5th International Semantic Web Conference,* pages 927-934, Springer, 2006.

[12] Celma, O. and Herrera, P. A new approach to evaluating novel recommendations. In *RecSys `08:* pages 179-186, New York, 2008.

[13] Celma, O. and Cano, P. From hits to niches?: or how popular artists can bias music recommendation and discovery. In *NETFLIX '08*: *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, pages 1-8, New York, NY, USA, 2008.

[14] Celma, O. *Music Recommendation and Discovery in the Long Tail.* PhD thesis.

[15] Pampalk, E. and Goto, M. Musicrainbow: A new user interface to discover artists using audio-based similarity and web-based labeling. *ISMIR*, pages 367-370, 2006.

[16] Duda, R. O. and Hart, P. E. *Pattern classification and scene analysis.* New York, 1973.

[17] Stone, M. Cross-validatory choice and assessment of statistical predictions. *Roy. Stat. Soc.,* 36:111-147, 1974.

[18] Herlocker, J. L., Konstan, J. A., and Riedl, J. T. Evaluating collaborative filtering recommendations. In *Computer Supported Cooperative Work*, pages 241-250, 2000.

[19] Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. Collaborative filtering recommender systems, 2007.

[20] Singha, S., Rashmi, K. S., and Sinha, R. Beyond algorithms: An HCI perspective on recommender systems, 2001

Music Recommendation and the Long Tail

Mark Levy
Last.fm
Karen House
1-11 Baches Street, London N1 6DL, UK
mark@last.fm

Klaas Bosteels
Last.fm
Karen House
1-11 Baches Street, London N1 6DL, UK
klaas@last.fm

ABSTRACT

Using a dataset of 7 billion recent submissions to the Last.fm Scrobble API[1], we investigate possible popularity bias in Last.fm's recommendations and streaming radio services. In particular we compare the recent listening of users who listen regularly to Last.fm streaming services with those who listen less often or never. Finally we describe a new service explicitly designed to make recommendations from the long tail, and analyse popularity effects across the recommendations which it suggests.

1. INTRODUCTION

Music lovers today have access to a previously undreamed of quantity and variety of recordings. Music is available through an increasing number of digital channels, including free online streaming services, "all you can eat" subscription services, and paid downloads, not to mention via illegal downloading and more traditional physical media. In one well publicised view [2], this proliferation in availability should lead to a reduction in the dominance of hits in our musical culture. With the development of advanced tools for search and recommendation, we should expect to see listeners discovering and enjoying a huge range of music that may be less popular overall, sitting somewhere in the so-called *Long Tail* of sales ranks, but which offers a good match for their own personal tastes.

The original long tail speculation was that it would become increasingly profitable to "sell less of more" by making large numbers of niche items easily available. Empirical studies of consumer behaviour suggest that this is indeed true, provided that enough choice is available, and that effective search and recommendation systems are provided to help users find their way around large inventories [4, 3]. A large recent study of consumer preference data, including user ratings for movies and music, shows that while not all users consume items in the long tail, "the vast majority of users are a little bit eccentric, consuming niche products at least some of the time", in particular reporting high average ratings for niche music [9].

Two lines of research suggest, however, that the utopian vision in which niche movies and music increasingly usurp the dominance of hits may not be borne out in practice. A recent study of the Netflix catalogue of movies shows that, on the contrary, demand for hits appears to rise, while that for niche products falls, as the number of available titles increases [12]. Meanwhile hit products continue to dominate the consumption of movies and music even for users who regularly explore the long tail [6]. Secondly, a number of studies of the very recommender systems which are supposed to support discovery in the long tail suggest that such systems are frequently prone to *popularity bias*, recommending globally popular items ahead of niche products [5, 7, 1].

In this paper we present an empirical study of the recommendations actually made by the widely-used Last.fm music recommender system, in particular via its streaming radio service, and set them in the context of wider music listening. As well as assessing the degree of popularity bias in these recommendations, we also compare the listening habits of a large group of music lovers regularly exposed to Last.fm's streaming radio with those of a second group who have no exposure to it. Finally we outline the design of a recommender system expressly designed to make recommendations from the long tail, and assess the popularity bias of a sample of the recommendations it produces.

The remainder of this paper is organised as follows: Section 2 briefly reviews previous work on popularity bias in recommender systems; Section 3 describes the data used as the basis for this study; Section 4 investigate the presence of popularity bias in Last.fm's radio streams, and Section 5 attempts to uncover any corresponding influence on users' wider listening habits; Section 6 outlines a music recommender explicitly designed to make recommendations in the long tail, and Section 7 draws conclusions.

2. PREVIOUS WORK

Three recent studies identify potential bias in recommender systems, particularly those based on *collaborative filtering* (CF). In [1] recommendations are generated using various well-established CF algorithms based on movie ratings from the MovieLens and Netflix datasets[2]. Over 84% of the MovieLens recommendations were for movies in the top 20% by

[1] http://www.last.fm/api/submissions

[2] http://www.grouplens.org, http://www.netflixprize.com

WOMRAD 2010 Workshop on Music Recommendation and Discovery, colocated with ACM RecSys 2010 (Barcelona, SPAIN)

number of ratings. No comparable figure is given for the Netflix recommendations, but the authors suggest that in both cases large gains in the diversity of recommendations can be achieved, with little cost to relevance, by suitable reranking techniques applied to the CF output.

The effect of recommendations on user behaviour is studied in a completely simulated setting in [7]. In the simulation users receive and consume recommendations from a CF recommender over a series of timesteps, so that over time the recommendations they receive are influenced by the previously recommended items which were added to their profile in previous rounds. The simulation was run repeatedly, with differing outcomes, but led in the great majority of runs to a decrease in the overall diversity of consumption.

CF recommendations are studied indirectly in [5], which considers the network defined by Last.fm's similar artist relationships. These relationships provide one of the sources of data used in Last.fm's recommender system, and can also be directly navigated as links on the Last.fm website, providing an active form of music discovery. Besides observing that Last.fm's similar artist lists are dominated by other artists with a similar level of popularity, [5] computes various network metrics to support the assertion that "CF tends to reinforce popular artists, at the expense of discarding less-known music", essentially by showing that navigation from popular to long tail artists often involves traversing a large number of artist links.

While all three of these papers discuss the effects of CF recommender systems, none of them considers a dataset of real recommendations made by a deployed system. In this paper we use Last.fm submissions data, defined fully in the next Section, to study the effect of a large-scale recommender system in practice.

3. DATA

Last.fm allows music lovers to *scrobble* details of their music listening. Scrobbling is available from media players and streaming services either though native support or via a suitable plugin, and is built in to some hardware devices. The Scrobble API[3] supports the submission of various events: in this paper we distinguish between *radio listens*, which record the act of playing a track via one of Last.fm's own streaming radio stations, and *scrobbles*, where the track played comes from any source other than Last.fm. In both cases the submitted metadata includes an artist name: for a scrobble this is typically drawn from the ID3 tags of the track being played.

Last.fm provides various types of streams, including *Similar Artists* and *Tag* radio, launched by supplying a seed artist or tag respectively, available to anyone, and *Recommendation* and *Library* radio, available to any user registered for scrobbling. Recommendation radio plays tracks by artists selected for the user by Last.fm's recommender, while Library radio plays tracks by artists previously scrobbled by the user. Users typically listen to Last.fm's radio stations through the flash player on the Last.fm website, or via a client program on their computer, phone or games console. In each case, information is displayed about the artist of the current track, including links to the artist's page on the Last.fm website, lists of similar artists, etc. While Recommendation radio is clearly an explicit recommendation service, all the stations can be considered as offering implicit recommendations, with Similar Artists radio in particular relying on underlying similarity data which also forms part of the input to Last.fm's recommender system. Even Library radio can be regarded as providing a form of non-novel recommendation, as it may remind the user of artists whom they like but have not listened to for some time.

[3]http://www.last.fm/api/submissions

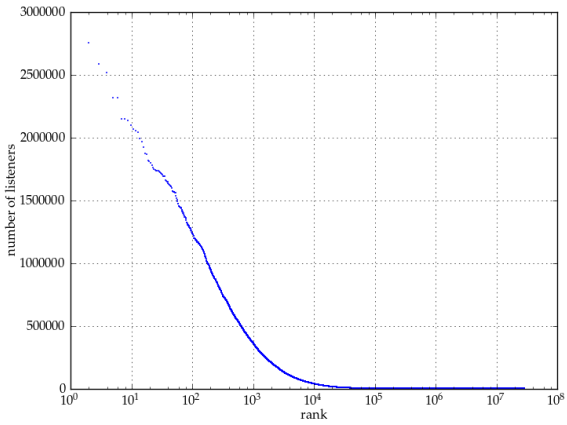


Figure 1: Artist popularity amongst Last.fm users.

In the following analysis we therefore pay special attention to Recommendation radio, but also consider the influence of Last.fm streaming radio as a whole. For the time being we neglect the influence of the recommendations displayed on users' Last.fm home pages and dedicated recommendation pages. The dataset used consists of over 7 billion submissions to the Scrobble API received between January and May 2010.

4. POPULARITY BIAS

The most widely-used measure of the diversity or, conversely, *concentration* of a set of products consumed by a group of users is the Gini coefficient [8], and this has also been applied to measure popularity bias within recommendations [7]. The Gini coefficient is computed from the area bounded by the Lorenz curve, which, in the case of artist recommendations, plots the proportion of the total number of recommendations made cumulatively for the bottom $x\%$ of artists recommended. The Gini coefficient is not ideal for our purposes here, as it depends on artist ranks within the set of recommendations being evaluated, i.e. it would show high concentration for a recommender that overwhelmingly recommended a small number of artists, even if all the artists it recommended belonged to the long tail. In the Sections that follow we therefore show plots similar to Lorenz curves, but showing the cumulative proportion of recommendations made in relation to artist ranks according to their *global* popularity, based simply on the overall total number of scrobbles received at the time of writing, shown in Fig. 1. We can also use this data to define what we mean by a "long tail" artist. Fitting Kilkki's informal model [10] suggests that this is any artist below rank 20,000; Fig. 1 shows, however, that in reality popularity flatlines slightly further down the tail, and that a reasonable definition of a long tail artist is one at rank 50,000 or below.

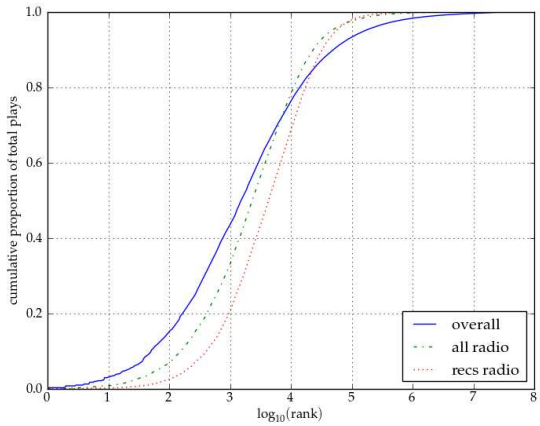


Figure 2: Popularity bias in Last.fm radio. Cumulative plays by artist rank for Recommendation radio and for all Last.fm radio stations. For comparison we also show the cumulative proportion of all scrobbles received during the same period.

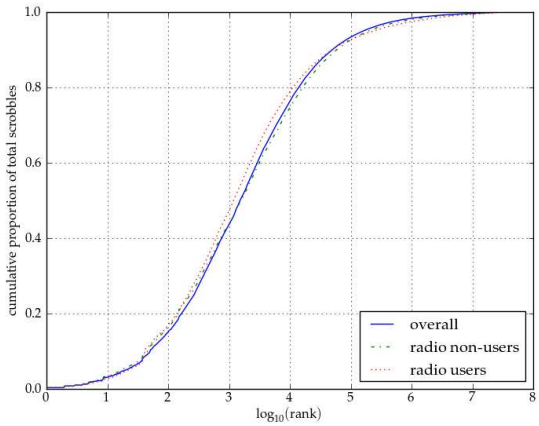


Figure 3: Possible influence of Last.fm radio. The plots show the cumulative proportion of scrobbles received by artist rank for two groups of users, one regularly exposed to Last.fm radio and the other completely unexposed to it.

Fig. 2 shows the distribution of ranks for artists played on Last.fm Recommendation radio, and on all Last.fm radio stations, compared with the distribution for all tracks scrobbled in same period. We observe that Last.fm radio is somewhat biased away from hit artists in comparison to the listening of Last.fm users as a whole, while Last.fm's recommendations are even more strongly biased towards lower ranking artists. In particular we see that artists in the top-1000 of overall listening make up 40% of scrobbles but only 20% of plays on Recommendation radio. Recommendation radio plays the same proportion of long tail artists as are listened to overall, but includes fewer plays of the lowest ranked artists: it is reasonable to assume, however, that artists scrobbled at those ranks include many whose tracks are not readily available for streaming, as well as spurious artists based on submissions with incorrect metadata that is not repaired by Last.fm's automatic correction system.

5. INFLUENCE

To expose the possible influence of Last.fm recommendations on users, we first create a set of active listeners by taking all users who registered during the five months under consideration and then scrobbled at least 500 but no more than 20,000 tracks during that time. The upper limit removes spammers and other technically-minded enthusiasts whose scrobbles represent a superhuman quantity of listening within that period, while the lower limit ensures that we have a reasonable amount of listening data for all of the users under consideration. We then draw two samples from this set of listeners. The first contains all users who had no exposure to any Last.fm radio station within the period (or indeed at any stage, as we include only newly-registered users). The second group contains all users for whom radio listens made up 25-75% of their submissions, i.e. these listeners are highly exposed to Last.fm radio, but also make a significant number of scrobbles for listening outside Last.fm.

Fig. 3 shows the distribution of artists scrobbled by each of these groups in the first five months of 2010, again compared with that for all tracks scrobbled during the same period. We observe a bias towards more popular artists in the mid region for the group of radio listeners, but it is small compared with the biases in artist popularity for radio plays shown in Fig. 1, and, more importantly, clearly not correlated with them. To control for demographic or other systematic differences between users who listen frequently to radio and those who never do so, in Fig. 4 we compare scrobbles for users with low exposure to radio, making up 10-50% of their scrobbles, to those with radio making up 50-90% of their scrobbles. In contrast to Fig. 3, this shows a slight bias towards the long tail in users with higher exposure to radio. We can conclude that there is no evidence that radio and recommendations cause a systematic bias towards more popular artists.

6. LONG TAIL RECOMMENDATIONS

We build a prototype recommender for long tail artists using conventional item-based CF. We first identify a suitable candidate pool of long tail artists from which to draw our recommendations. For each artist in our overall catalogue we then find the most similar k artists within the pool, based on scores computed by comparing both scrobbles and tags applied to each artist. When a user u requests recommendations, we create a profile of artist weights W_u based on their scrobbles, and build up a candidate set containing the top-k similar artists in the pool for each artist in W_u. We then score each candidate artist a based on their similarity to artists in the user's profile, computing a score $P_{u,a}$ using the well-known weighted sum method [11], finally returning the top-N highest scoring artists:

$$P_{u,a} = \sum_{a' \in W_u} sim(a, a') w_u(a') \quad (1)$$

where $sim(a, a')$ is the similarity between a and a', and $w_u(a')$ is the weight assigned to a' in the the user profile.

To obtain a suitable pool of long tail artists, we start with all artists with tracks currently available in the Last.fm "Play direct from artist" scheme, under which unsigned artists or labels holding suitable rights can make tracks available for

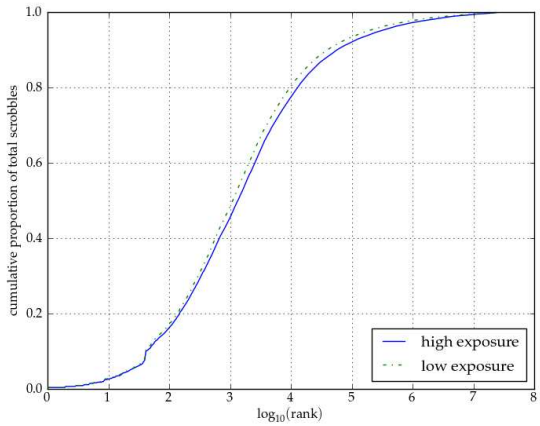


Figure 4: Scrobbles received for users with low and high exposure to Last.fm radio respectively.

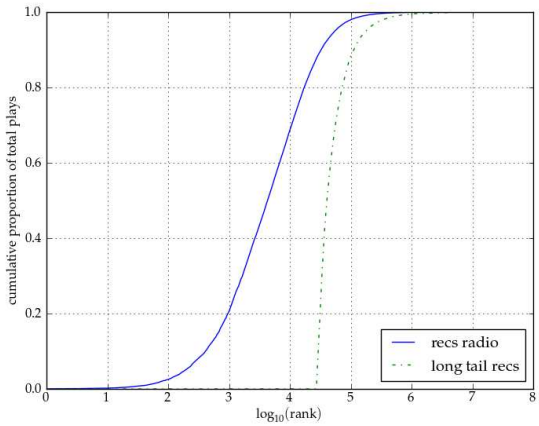


Figure 5: Long tail recommendations vs plays on the main Recommendation radio.

free streaming from their Last.fm pages. This scheme is aimed at niche and new artists, but to be sure that artists in the pool are indeed from the long tail, we also apply a hard cutoff on current overall reach, removing any artists with more than 10,000 listeners. Finally to mitigate problems with artist disambiguation in the long tail, where new or niche artists have the same names as more popular artists, we mine Last.fm wiki entries for key phrases indicating multiple artists with the same name, removing any affected artists from the pool. The resulting set of long tail artists is updated daily, but at the time of writing contains 118,000 artists.

To study the popularity distribution amongst artists suggested by this new system, we generate 50 recommendations for each of a sample of 100,000 active Last.fm users, defined as users who have visited the Last.fm website within the last week. Fig. 5 shows the resulting distribution, compared to that for plays on the main Recommendation radio station during the first five months of 2010. Approximately 90% of the sampled recommendations are for artists in the mid to long tail, with ranks 25,000 to 100,000, with the remaining 10% being for the lowest ranking artists. While the previous Section suggests that the influence of recommendations may be limited, we can reasonably hope that the prototype recommender will gradually stimulate increased interest in the long tail.

7. CONCLUSIONS

A comparative analysis of artists chosen by Last.fm's recommender system and a large body of listening data suggests that, contrary to claims in the literature based on laboratory experiments, real world music recommenders do not necessarily exhibit strong popularity bias. Our results suggest that, in any event, the influence of such a recommender on users' general listening may be limited. Finally we sketch the design of a prototype recommender designed explicitly to suggest artists from the long tail. Future work includes a user evaluation of the prototype system, which is now publicly available[4].

[4]http://playground.last.fm/demo/directrecs

8. REFERENCES

[1] G. Adomavicius and Y. Kwon. Toward more diverse recommendations: Item re-ranking methods for recommender systems. In *Proc. 19th Workshop on Information Technologies and Systems*, 2009.

[2] C. Anderson. *The Long Tail. Why the future of business is selling less of more.* Hyperion, 2006.

[3] E. Brynjolfsson, Y. Hu, and D. Simester. Goodbye pareto principle, hello long tail: The effect of search costs on the concentration of product sales. Technical report, MIT Center for Digital Business, 2007.

[4] E. Brynjolfsson, Y. Hu, and M. Smith. From niches to riches: anatomy of the long tail. *Sloan Management Review*, 47(4):67–71, 2006.

[5] O. Celma and P. Cano. From hits to niches? or how popular artists can bias music recommendation and discovery. In *Proc. 2nd Netflix-KDD Workshop*, 2008.

[6] A. Elberse. A taste for obscurity? an individual-level examination of "Long Tail" consumption. Technical report, Harvard Business School, 2007.

[7] D. Fleder and K. Hosanagar. Recommender systems and their impact on sales diversity. In *EC '07: Proceedings of the 8th ACM conference on Electronic commerce*, 2007.

[8] C. Gini. Measurement of inequality of incomes. *The Economic Journal*, 21(121):124–6, 1921.

[9] S. Goel, A. Broder, E. Gabrilovich, and B. Pang. Anatomy of the long tail: ordinary people with extraordinary tastes. In *WSDM*, 2010.

[10] K. Kilkki. A practical model for analyzing long tails. *First Monday*, 12(5), 2007.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, 2001.

[12] T. Tan and S. Netessine. Is Tom Cruise threatened? using Netflix Prize data to examine the Long Tail of electronic commerce. Technical report, Wharton Business School, University of Pennsylvania, 2009.