



UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
DIPARTIMENTO DI TECNOLOGIE DELL'INFORMAZIONE

SCUOLA DI DOTTORATO IN INFORMATICA
Settore disciplinare INF/01

TESI DI DOTTORATO DI RICERCA
CICLO XXIII

**SERENDIPITOUS MENTORSHIP IN MUSIC
RECOMMENDER SYSTEMS**

Eugenio Tacchini

Relatore: Prof. Ernesto Damiani
Direttore della Scuola di Dottorato: Prof. Ernesto Damiani

Anno Accademico 2010/2011

Acknowledgements

I would like to thank all the people who helped me during my Ph.D.

First of all I would like to thank Prof. Ernesto Damiani, my advisor, not only for his support and the knowledge he imparted to me but also for his capacity of understanding my needs and for having let me follow my passions; thanks also to all the other people of the SESAR Lab, in particular to Paolo Ceravolo and Gabriele Gianini.

Thanks to Prof. Domenico Ferrari, who gave me the possibility to work in an inspiring context after my graduation, helping me to understand the direction I had to take.

Thanks to Prof. Ken Goldberg for having hosted me in his laboratory, the Berkeley Laboratory for Automation Science and Engineering at the University of California, Berkeley, a place where I learnt a lot; thanks also to all the people of the research group and in particular to Dmitry Berenson and Timmy Siau for the very fruitful discussions about clustering, path searching and other aspects of my work.

Thanks to all the people who accepted to review my work: Prof. Richard Chbeir, Prof. Ken Goldberg, Prof. Przemysław Kazienko, Prof. Ronald Maier and Prof. Robert Tolksdorf.

Thanks to 7digital, our media partner for the experimental test, and in particular to Filip Denker.

Thanks to Letizia Guagnini for the graphic design of the interface of Mentor.FM and to Simone Magnaschi for the HTML/CSS/JS conversion of the design.

Thanks to Filippo Losi, Leonora Fortunati and Camilla Sampò, who helped to understand better which are the different communities behind some of the musical worlds found.

Thanks to the people of stackoverflow.com for their willingness to help others using their impressive skills, thanks in particular to Itamar Katz for the precious suggestions received about the optimization of the the boolean similarity computation.

Thanks to Marco della Vedova for the suggestions and the fruitful discussions.

Thanks to all the friends who inspired and supported me during these years.

Finally, thanks a lot to my family for supporting me over the years and thanks again to my father who, when I was 8 years old, bought for me a Commodore home computer, by mistake. Considering that computer science became for me first a passion and then a job, I would say it was an excellent example of serendipity.

Abstract

Nowadays the amount of content and products easily available on-line for purchase or fruition is so high that recommender systems represent an important resource for users in order to get suggestions about items (songs, movies, books, news, products in general...) they might like.

For many years, research, in the field of recommender systems focused on improving accuracy, i.e. improving the precision with which the systems predict the rate that a given user would give to a given item.

In the last years, an increasing number of efforts have been directed towards other important aspects such as novelty, diversity and serendipity of recommendations. In particular, with serendipity, in this context, we refer to the ability of a recommender system to propose unexpected and liked recommendations. Serendipity is likely the aspect which has received the least attention and it is the one, in this work, we focus more on.

The aim of this thesis is to propose techniques which can be adopted by recommender system designers in order to increase serendipity while keeping an acceptable level of precision of the recommendations. We work in the domain of music, which presents a particularly suitable context for trying to propose non-obvious recommendations, mainly due to the lower cost, respect to other domains, of “bad” recommendations (listening to a song a user dislikes is not much time consuming).

The work proposes a collaborative-filtering method to classify artists, based on the Affinity Propagation clustering algorithm and on listening logs as data source. The classification, together with a list of the artists a user likes, is used to detect which musical clusters (called “musical worlds”) the user is not familiar with. A technique to synthetically represent each cluster, based on freely chosen keywords (folksonomy), is also presented.

A novel recommendation method based on gradual exposure and on a variation of the user-based collaborative filtering approach is proposed. The said method exploits the knowledge of the most eclectic users (we decided to call them “mentors”) to choose, from the unfamiliar musical clusters, the ones which are more likely to contain serendipitous music for the active user.

Once a target musical cluster has been chosen, a playlist is created, which starts with songs by artists who tend to be borderline in respect to the user's taste and continues with songs by artists who tend to be, gradually, closer to the most representative artist of the target cluster.

A real music recommendation radio has been developed, implementing the techniques proposed and a traditional top-10 item-based recommender. The radio has been used as a validation test, considering the traditional recommender as a baseline to define which recommendations were expected and which ones were unexpected. The test session suggested that the proposed approach overcomes a method which relies on randomness in terms of a novel measure, called “serendipity cost” (measured as the total number of disliked songs over total number of serendipitous songs) and in term of cohesion, maintaining a “total cost” (measured as the total number of disliked songs over total number of liked songs, which can be considered an index of precision) which is much lower than the cost related to the random approach and closer to the cost of a traditional item-based recommender systems (1.03 for the method proposed, 0.46 for the traditional recommender, 2.77 for the random).

The method we proposed in order to choose and order the intermediate artists in a playlist, based on graph search techniques, is used to gradually

expose the user to the target musical world, following the intuition that showing a connection between the target musical world and the music the user is closer to can help him to accept the (unexpected) recommendation. This method, however, can itself be considered an achievement of this work and applied not only in this context but anytime the automatic production of a playlist, having in input the first artist, the last artist and a cohesion (distance in a playlist between an artist and the following one¹) constraint, is needed.

¹ Note that cohesion in literature is usually defined as the average distance in a playlist between a song and the following one so in this sentence the term is used in a broader sense

Contents

1 Introduction.....	1
1.1 Recommender Systems.....	1
1.1.1 Definition.....	1
1.1.2 Aims.....	2
1.1.3 Data sources.....	4
1.1.4 Techniques.....	4
1.1.4.1 Content-based.....	5
1.1.4.2 Collaborative-filtering based.....	6
1.1.4.2.1 User-based.....	6
1.1.4.2.2 Item-based.....	11
1.1.4.3 Hybrid.....	12
1.1.4.4 A comparison between content-based and collaborative-filtering-based techniques.....	12
1.2 Recommender Systems in the Music Domain.....	14
1.2.1 Peculiarities.....	14
1.2.1.1 Huge items space.....	14
1.2.1.2 Low cost/consumption time.....	14
1.2.1.3 Very high per-item reuse.....	15
1.2.1.4 Contextual and mood usage.....	15
1.2.1.5 Consumed in sequence.....	16
1.2.1.6 Implicit feedback evaluation.....	16
1.2.2 Applications.....	17
1.2.2.1 Pandora.....	17
1.2.2.2 Last.fm.....	18
1.2.2.3 Musicoverly.....	18
1.2.2.4 The Echo Nest.....	19
2 Motivations, goals and state of the art.....	20
2.1 Overspecialisation problem.....	20
2.2 Diversity, novelty, serendipity and user satisfaction.....	21

2.3 Problem statement	26
2.4 Related work.....	27
3 Proposed approach.....	34
3.1 Dataset.....	34
3.2 Similarity measures.....	38
3.2.1 Introduction.....	38
3.2.2 Boolean similarity.....	40
3.2.3 Pearson Correlation similarity.....	42
3.2.4 Folksonomy-based similarities.....	46
3.2.4.1 Introduction to folksonomies.....	46
3.2.4.2 Folksonomies in the music domain.....	46
3.2.4.3 Music folksonomies and Music Information Retrieval / Recommendations.....	48
3.2.4.4 Similarity measures.....	52
3.3 Artists' classification.....	54
3.3.1 Introduction.....	54
3.3.2 Affinity Propagation clustering and Musical Worlds detection..	55
3.3.2.1 The Affinity Propagation algorithm.....	55
3.3.2.2 Application to our Artists' classification problem.....	58
3.3.2.3 How to represent a Musical World.....	62
3.3.2.4 Examples and first discussion.....	62
3.4 Eclecticism level evaluation.....	67
3.5 “Mentor” approach.....	74
3.6 Playlist generation.....	76
3.6.1 Introduction.....	76
3.6.2 Playlist generation, given in input the artists.....	77
3.6.3 Playlist generation, given the first and the last artist.....	81
3.6.3.1 Introduction.....	81
3.6.3.2 The Artists' graph.....	81
3.6.3.3 Application of the Floyd's Algorithm.....	82
3.6.3.4 Examples.....	84
4 Application: mentor.fm.....	96
4.1 Introduction.....	96
4.2 Interface and features.....	96
4.3 Technologies.....	98
4.4 Playlists creation and modalities.....	103
4.4.1 Normal playlist, random profile.....	103
4.4.2 Normal playlist, dj profile.....	104
4.4.3 Surprise me! playlist, random profile.....	104
4.4.4 Surprise me! playlist, mentor profile	104
4.5 Assignment of user profiles.....	105
5 Evaluation results.....	107
5.1 Evaluation of Artists' Classification.....	107
5.1.1 Metrics.....	107
5.1.2 Results.....	109

5.2 Evaluation of the surprising suggestions.....	110
5.2.1 Metrics.....	110
5.2.2 Results.....	112
5.2.3 Discussion.....	116
5.3 Evaluation of the cohesion.....	117
5.3.1 Metrics.....	117
5.3.2 Results.....	118
5.3.3 Discussion.....	119
6 Conclusions, limitations and further research.....	120
6.1 Contributions.....	120
6.2 Limitations and future research.....	122
References.....	125
Appendix A.....	132
Appendix B.....	134

1

Introduction

1.1 Recommender Systems

1.1.1 Definition

Recommender systems are personalized information agents that provide recommendations: suggestions for items likely to be of use to a user [Burke, 2007].

Items is a generic term used in literature to denote the object of a recommendation, recommender systems can in fact work on very different kind of items, for example they can suggest:

- which books to read;
- which music to listen;
- which movies to watch;
- which friends to add in a social network.

Usually, a recommender system focus on a specific kind of item and is designed and developed to effectively work on that [Ricci et al., 2010].

Recommender systems emerged as a research area in the 1990s and the interest in the field has increased in the recent years, also because of the

grow of e-commerce web sites which has multiplied the options available to the users, making harder the process of choosing² [Ricci et al., 2010].

1.1.2 Aims

Recommender systems are used by the final users for different kind of tasks, we summarize here the most common ones, considering the analysis provided by [Herlocker et al., 2004] and [Ricci et al., 2010].

- Find some good items. This is probably the most common task: users ask recommender systems to extract automatically, from a collection of items, the subset of the them which is the most interesting (according to the recommender system predictor algorithm) for them. This problem is often called *top-n*, because just the first n (e.g. 10) most important items are returned. An example could be a movie recommender systems which recommend to a user, according to his profile, the top 10 movies he would enjoy more.
- Find all good items. Sometimes users need recommendations to extract all the items which can be considered interesting for a particular need: imagine for example a prior art search application, which helps a user, according to a topic, to find all the related patents, publications and public discussions, extracting them from an archive. While the difference between the first and the second task could seem trivial, according to the task the design of the system can change a lot: for the first task the system should above all minimize the number of *false positives*, preferring *accuracy* to *coverage*, for the second task the system should, above all, minimize the number of *false negative*, preferring coverage to accuracy. Let's consider the prior art search example described above: in that scenario, a user can accept a small number of false positive (items the system thinks are

² For some work about the relation between availability of choice and user's benefit see [Schwartz, 2004]

relevant even if they are not) but he would probably hardly accept false negative errors because they would prevent him from satisfying his needs (i.e. find all good items).

- Bundle recommendation. In this scenario the recommender system suggests a group of items. For example a desktop computers e-commerce Web site could provide, through a recommendation engine, suggestions about sets of computer components (motherboard, video card, hard disk, etc.) which can work well together, according to the preferences of the user on the single items.
- Sequence recommendation. In this scenario the user asks for an ordered sequence of items and the order of the items does matter. An example could be a music playlist recommender system or a system which recommends a sequence of publications to gradually introduce a user to a topic, depending on the confidence level he has.
- Group recommendation. In this scenario the recommender system provides recommendation not for a single user but for a group of people, trying to aggregate the profiles of all the users belonging to the group. For example a movie recommender system could provide suggestions for a movie to a group of friends who weekly meet to watch TV together.
- Annotation in context. This is the first recommender system scenario and, even if now it is probably not anymore a very common task for the typical recommender application, we present it to give a wider overview. For this kind of uses, the system presents to a user both the “good” and the “bad” items, annotating them according to the relevance for the user. An example could be a news reader which shows all the available items to the users, labeling them with

different colours according to how much the user, presumably, is interested.

1.1.3 Data sources

Data used by recommender systems refers to three kind of objects: items, users and transactions [Ricci et al., 2010].

Items can be represented in several ways, depending on the recommendation technique used by the system; for example collaborative-filtering-based recommender systems (see 1.1.4.2) represent items using the ratings the users gave to them, while content-based recommender systems (see 1.1.4.1) represent items using some attributes whose values depend on the characteristics of each item.

Also the representation of users depends on the technique used by the recommender system, different kinds of user models can be used, in fact, to describe a user and give, accordingly, personalized recommendations; while a collaborative-filtering-based recommender system simply represents users through the ratings they assigned to items, other approaches could model users according to demographic data or other kind of knowledge.

Transactions, finally, are interactions between users and items; the typical transaction recorded by recommender systems is a rating that a user assigned to an item, the rating can be explicit (e.g. a numeric score in the 1...5 range) or implicit: an evaluation derived from the user's behaviour (e.g. if a user rented several time movies by a specific director, a high rating for that director can be assumed).

1.1.4 Techniques

Recommender systems are usually classified into three main categories: content-based recommender systems, collaborative filtering-based recommender systems and hybrid recommender systems [Adomavicius and Tuzhilin, 2005].

1.1.4.1 Content-based

Content-based recommender systems rely on the analysis of the items to suggest: each item (e.g. a document, a song, a movie...) is analyzed by feature extraction techniques in order to represent its content in a specific information space; a document, for example, can be described through a keywords vector. Typically, a profile of the user is created considering the items he rated in the past, and the recommendation process consists in comparing the user profiles against the representation of the available items in order to suggest other items similar to ones the user liked in the past [Ricci et al., 2010].

In order to create the profile of the user, two different techniques can be used: explicit feedback and implicit feedback. The explicit feedback technique requires the user to evaluate the items, while the implicit feedback technique infers the evaluation according to the activity of the user; for example, if a user purchases an item or repeatedly listen to a song, the system can infer a positive feedback on the item or on the song [Ricci et al., 2010].

Different strategies can be adopted to get explicit feedback, the most used are:

- like/dislike: the users classify items using a binary rating scale;
- ratings: the user classify items using a numeric scale, for example rating items using a 1...5 scale;
- text comments: the user can give a descriptive feedback writing a text comment about an item [Ricci et al., 2010].

The feedback techniques described are not just related to content-based recommender systems and are in fact also used with other type of recommender systems.

1.1.4.2 Collaborative-filtering based

Collaborative filtering (CF) is the process of filtering or evaluating items through the opinions of other people [Schafer et al., 2007]. Typically, these systems work on a users-items ratings matrix (where for each pair of user-item, the matrix provides, if available, the rating the user gave to the item) trying to estimate the ratings for the user-item pairs not yet available.

In details, the main idea behind collaborative filtering techniques is that the rating of a user u_x for an item i_k should be similar to the one another user u_y gave to the same item if u_x and u_y are similar i.e. if they rated similarly other items; following the same assumption, from a different perspective, the rating of a user u_x for two items i_k and i_z should be similar if the two items are similar i.e. if other users rated similarly these two items [Ricci et al., 2010]. For some classic work on collaborative-filtering see [Breese et al., 1998], [Sarwar et al., 2001] and [Goldberg et al., 2001]

Collaborative filtering techniques can be classified in two categories: memory-based and model-based [Breese et al., 1998]. In memory-based collaborative filtering recommender systems, the ratings stored in the system are directly used to predict ratings for user-item pairs which are still not available in the system; model-based techniques, instead, use the ratings stored in the system to produce a predictive model and use the model to predict the ratings for user-item pairs not available: some examples of model-based approaches are: Bayesian Clustering, Latent Semantic Analysis or Singular Value Decomposition [Ricci et al., 2010]. Memory-based approaches are furthermore classified in two different categories: users-based and items-based [Candillier et al., 2007, Ricci et al., 2010]; we will present in details these two approaches in the following two paragraphs.

1.1.4.2.1 User-based

The user-based approach predicts the rating that a user u_i will give to an item i_x according to the ratings that other users, similar to u_i (and called

“neighbours”) gave to the same item. The similarity between the users (i.e. the neighbours set computation) is based on the ratings the users gave to others items [Ricci et al., 2010].

In figure 1, 2 and 3 a toy example of user-based collaborative filtering recommender system is presented: in our hypothetical movie recommender system, there are five users and four movies and each user can rate each of the movie using a discrete numerical value in the range 1...5, where 1 means the user does not like the movie and 5 means the user likes the movie very much; the recommender system tries to predict the ratings for the (user, item) pairs which still do not have a rating associated and proposes to the user the relative item if the predicted rating is 4 or 5. In the instance presented, all the users rated all the movies except from Tom, who rated only three movies out of four. We want to predict the rating that Tom would give to the movie “Lost in translation”, in order to understand if we should recommend it to the user or not. This can be accomplished in two steps:

1. Select Tom's neighbourhood (see fig. 1)
2. Compute the rating according to the neighbourhood's rating (see fig. 2)

Tom's neighbours are the users who, for the movies “2001: A space Odyssey”, “Seven Samurai” and “Fight Club”, gave ratings similar to Tom's: in our example Bob and Anna. A simple method to predict the rating for the (Tom, “Lost in translation”) pair is to compute the mean between Bob's and Anna's rating on “Lost in translation”: $(5+3)/2 = 4$; the movie is then suggested to Tom.

1 - Introduction

<i>users</i> ↓	2001: A Space Odyssey	Lost in translation	Seven samurai	Fight club	← <i>movies</i>
John	1	3	3	2	
Bob	4	5	5	4	
Alice	2	2	1	2	
→ Tom	5	??????	5	4	
Anna	3	3	5	4	

↑
ratings, 1...5

Figure 1: User-based approach, the ratings matrix

<i>users</i> ↓	2001: A Space Odyssey	Lost in translation	Seven samurai	Fight club	← <i>movies</i>
John	1	3	3	2	
Bob	4	5	5	4	←
Alice	2	2	1	2	
→ Tom	5	??????	5	4	
Anna	3	3	5	4	←

↑
ratings, 1...5

neighbours

Figure 2: User-based approach, step 1

USERS	2001: A Space Odyssey	Lost in translation	Seven samurai	Fight club	← movies
John	1	3	3	2	
Bob	4	5			
Alice	2	2			
→ Tom	5	??????	5	4	neighbours
Anna	3	3	5	4	

ratings, 1...5

(5+3)/2=4
suggest!

Figure 3: User-based approach, step 2

The example presented is a simplification of the user-based collaborative filtering technique; in the followings we will add some details and present some of the typical issues that a real-world application faces.

1. Sparsity and cold-start problem. In order to work properly, a collaborative filtering algorithm needs a certain amount of data: if the ratings matrix is too sparse, the accuracy of the suggestions can decrease because having just a few ratings per user makes the users' similarity computation unreliable and having just a few ratings per item makes the items ratings prediction unreliable. In particular, this happens frequently for new users and new items: if a new user joins the system, he will probably not get reliable suggestions until he rates a certain number of items; if a new item is introduced in the system it will probably not got reasonably recommended until a certain number of users rate it; this is called in literature “cold start problem” [Schein et al., 2002].
2. Ratings normalization. Each user has different rating habits: some users for example tend to give, on average, high ratings, while some others keep the higher ratings for exceptional cases. To keep into

account these different scales, a normalization process is often applied; two common techniques are *mean-centering* and *z-score normalization* [Ricci et al., 2010]. In the former, the mean of the ratings assigned by a user is subtracted to each of the rating; in the latter, in addition, each rating is also divided by the standard deviation of the relative user's ratings.

3. Similarity measures. Several measures have been proposed in literature to express the similarity between two users: a very common approach consists in considering each user as a vector of ratings and compute the correlation between the two vectors using for example the Pearson Correlation index, as shown in equation 1, where I and J are two vectors, \bar{I} and \bar{J} are the means of I and J and I_x/J_x is the x^{th} element of I/J .

$$PC(I, J) = \frac{\sum_{x=1}^n ((I_x - \bar{I}) \times (J_x - \bar{J}))}{\sqrt{\sum_{x=1}^n (I_x - \bar{I})^2 \times \sum_{x=1}^n (J_x - \bar{J})^2}} \quad (1)$$

4. Neighbours selection. Once the similarities between users have been computed, different approaches can be adopted to choose the neighbours; a popular approach considers as neighbours of a users the top-N similar users of the system, where N is a parameter chosen by the designer of the system; another approach, instead of considering a fixed number of neighbours, compute the neighbourhood of a user as the set of all the users having a similarity with him higher than a threshold.
5. Ratings prediction. After having chosen the neighbours, a common approach used to predict, for a user, an item's rating, consist in computing the mean of the ratings given by the user's

neighbourhood, weighted by the similarity between each neighbour and the user (more similar neighbours impact more on the prediction).

1.1.4.2.2 Item-based

The item-based approach works on the same idea of the user-based one, looking at the problem from a different perspective: in order to predict the rating that a user u_x would give to an item i_k , instead of computing the neighbourhood of the user u_x , it computes the neighbourhood of the item i_k (i.e. the similar items, the ones rated similarly by other users) and uses the ratings the user u_x gave to those items to predict the rating he would give to the item i_k .

All the details we gave about user-based techniques are valid, specularly, for item-based techniques, in details:

1. Sparsity and cold-start problems. These problems can affect the item-based approach as well.
2. Rating normalization. The same method described above can be applied to the item-based technique, in this case considering mean and standard deviation of an item (instead of the mean and the standard deviation of an user).
3. Similarity measures. The vectors are used, in this case, to represent the items.
4. Neighbours selection. Neighbours of items, instead of neighbours of users, are selected.
5. Ratings prediction. As anticipated, to predict, for a user, an item's rating, this approach uses the mean of the ratings given by the user to the item's neighbourhood, weighted by the similarity between each neighbour and the item (more similar items impact more on the prediction).

In order to choose between a user-based and an item-based approach, several aspects can be taken into consideration, where one of the most important one is the accuracy of the resulting system. In general, a user-based recommender system is preferable because more accurate when the number of items overcomes the number of users, while an item-based approach is more accurate when the number of users overcomes the number of items. The reason is that when the number of users overcomes the number of items, an item-based approach provides a problem instance where the neighbours are less (respect to a user-based approach) but have an higher number of common ratings and therefore gives a more confident prevision; specularly, when the number of items overcomes the number of user, a user-based approach provides a problem instance where the neighbours are less (respect to an item-based approach) but have an higher number of common ratings. Another aspect to take into consideration is that item-based recommenders usually provide safer recommendations, more in line with user's taste, while user-based recommenders are more likely to propose serendipitous recommendations [Ricci et al., 2010].

1.1.4.3 Hybrid

Hybrid recommender systems combine collaborative-filtering-based techniques with content-based techniques: for example a collaborative filtering recommender system can use a content-based approach when a new item is introduced in the system, in order to alleviate the cold-start problem.

1.1.4.4 A comparison between content-based and collaborative-filtering-based techniques

Considering part of the description provided by [Ricci et al., 2010] (pp 78-79), in this paragraph we present the most evident, in our opinion, advantages and disadvantages of the two recommender system techniques described above.

The main advantages of the content-based technique are the following:

- It does not suffer from the item cold-start problem: even a new (just introduced) item can be analyzed, described and recommended. It can, however, suffer from the user cold-start problem: if the system needs to create a user profile (based on a certain amount of ratings) before being able to propose reasonable recommendations, new users could get bad recommendations during the period that immediately follows the subscription.
- It can work even with a limited number of users or items.

The main disadvantage is the following:

- The content analysis process is not trivial, it is domain dependent, often requires domain knowledge and sometimes it is not easy to extract from the items enough relevant information to describe them.

The main advantages of the collaborative-filtering-based technique are the following:

- Relatively simple implementation.
- More domain independent than content-based: having a users-ratings matrix, the same techniques can be applied regardless of the domain. Domain knowledge can, however, help exploiting domain peculiarities and therefore providing better recommendations (for example see paragraph 1.2.1 for peculiarities in the music domain).
- Reflects the real preferences of people.

The main disadvantages are the following:

- It can suffer from the cold-start problem, both for new users and new items.

- The accuracy depends on the amount of rating data available (this might be seen as another aspect of the cold-start problem).

1.2 Recommender Systems in the Music Domain

1.2.1 Peculiarities

Music has some peculiarities which makes its domain unique and which should be taken into consideration when designing recommender systems algorithms. Some of the peculiarities we want to highlight have been clearly discussed in [Lamer, 2011] and will be proposed here.

1.2.1.1 Huge items space

If we consider the song as the item to recommend, the typical content space which a recommender system has to deal with counts millions of items. The Apple iTunes store³ provides more than ten millions songs. Other domains, such as books or movies, usually work with a much smaller items space, even three order of magnitude less. This property has big implications in the design of the algorithms: the computational effort needed to calculate similarities among million of songs is not trivial. Furthermore, if we consider a collaborative-filtering technique it is possible that the rating data about a song is not enough to compute similarities. For this reasons research approaches and commercial applications sometimes approximate the problem computing similarities among artists instead of doing it among songs.

1.2.1.2 Low cost/consumption time

The price of a song is usually an order of magnitude lower than the price of a movie or a book; can be two or three order of magnitude lower than the price of other products such a laptop recommended by a computer e-shop.

³ <http://www.apple.com/itunes/>

The consumption time is also lower: a typical song can be listened in three/four minutes while for example a book can take two weeks to be read. Also this property affects the design of a recommendation algorithm: being the cost and the consumption time per item so low, a recommender system can safely being designed to provide more risky recommendations (higher unexpected/secure recommendation ratio); if a song recommendation is not appreciated by the user, he wastes just a very low amount of money (or even zero, if we consider streaming services based on a flat monthly fee, which are becoming quite popular) and three-four minutes of his time.

Furthermore, a music recommender system needs to be designed to provide an high amount of recommendations in a limited time (think about a Web radio powered by a music recommender system).

1.2.1.3 Very high per-item reuse

Loved songs are typically listened more than once, even tens or hundreds of times. Sometimes can happen that a user wants to listen twice in a row the same song. This does not happen with other content types: while it is possible and not uncommon to watch a movie more than once, it is unlikely to watch it twice in a row or to watch it hundreds of times. Music recommender systems should take into consideration this peculiarity, providing a good mix between already-known songs and new songs, while in other fields recommender systems typically avoid proposing an already consumed item.

1.2.1.4 Contextual and mood usage

The way we consume music is affected, more than in any other domain, by the context we are in during the consumption. For example there is music particularly suitable as “background” while people are doing some other activities, there is music suitable for a party or music suitable for jogging activity. The usage changes according to the mood as well, for example

sometimes we need energetic music because we feel tired, or we change the kind of music according to our happiness or again, we are remembering our adolescence and want to listen something that in our mind is related to that period. Recommender systems should take context into consideration.

1.2.1.5 Consumed in sequence

A song, the typical item in a music recommender system, is usually not consumed singularly: people usually listen several songs in sequence grouped in a *playlist*. This introduces an additional problem for a music recommender: how to order the items in the sequence; a different order produces a different user experience and therefore could impact on the user satisfaction.

1.2.1.6 Implicit feedback evaluation

While in most of the other domains recommender systems often focus on explicit feedbacks evaluation (for example a rating from 1 to 5 given by a user to a proposed movie) in the music field the implicit feedback evaluation is very important, valuable and often available in huge amount: the play-logs of a music service, including the number of times a user plays each song, can be an important (and sometimes the main) information we can use as input.

The number of times a user listened to a song or to an artist can be indeed used as an implicit feedback on the song/artist: it is straightforward to assume that, in general, people listen more the artists and the songs they like more. Determining the exact relationship between the number of times a song/artist is played and the level of likeness is, however, not trivial: as an example, it is pretty clear that popular artists have more chance to be played, furthermore, the function which links the playcounter to the likeness is probably not linear and not strictly increasing monotonic: over a playcounter threshold it is safe to assume that the level of likeness does not

further increase. In [Celma, 2008] for example, a normalization method which maps playcounter values to ratings in the 1...5 range is proposed.

1.2.2 Applications

In the following paragraphs we briefly present four real-world music recommender applications, selected because of their popularity, the different techniques used and the amount of information publicly available on the techniques themselves: Pandora, Last.fm, Musicoverly and The Echo Nest. While Pandora and Musicoverly mainly adopt a content-based approach, Last.fm is focused on collaborative filtering techniques; The Echo Nest exploits both the solutions applying an hybrid approach.

1.2.2.1 Pandora

Pandora⁴ is a music recommender radio based on the Music Genome Project⁵. The radio accepts in input one or more artists or songs (“seed” artists and songs) and creates and plays a radio station based on artists and songs similar to the ones given as input. The users can give an additional direct feedback while listening to the songs: a “thumbs up” and a “thumbs down” button can be used to give a “like” or “dislike” feedback to the system.

The recommender engine uses a content-based technique: each song included in the Pandora's music archive is listened by music experts and described using hundreds of attributes (“genes”), for example: “Heavy Drums”, “Acoustic Guitar Solo”, “Repetitive Song Structure” or “Abstract Lyrics”. For each gene, a numerical value in the range 1...5 is used to specify at which extent that gene characterizes a song. The list of genes which should be used to describe a song is defined by the Music Genome Project and depends on the music genre of the song itself.

⁴ <http://www.pandora.com>

⁵ <http://www.pandora.com/mgp.shtml>

After having described each song as a vector of genes, the recommender system computes the similarities among songs calculating the distance between the vectors and uses the similarities to build the radio station according to the input received by the user.

Pandora has been founded in 2000 and it is available only in the United States of America.

1.2.2.2 Last.fm

Last.fm⁶ is a music Web service founded in 2002; the service creates a music profile of each user according to the songs the user listen to and provides, among the other features, a recommendation radio which mainly works using a collaborative-filtering approach. Since the dataset used in this thesis comes from Last.fm, we will discuss in detail the service in chapter 3.1 .

1.2.2.3 Musicoverly

Musicoverly⁷ is a personalized recommendation radio which provides recommendations using a content-based approach. Each of the song in the Musicoverly archive has been listened by music experts and described using 40 acoustic parameters⁸. The songs are then suggested and played to the users according to different strategies, where the main one is using as input a mood-selection tool, the “mood pad”. The mood pad is a pad where the user can set a mood according to two axes: the x axis, which ranges between “dark” and “positive” mood and the y axis, which ranges from “calm” to “energetic” mood. Since each song, according to the Musicoverly algorithm, can be projected into a specific position of the mood pad, the radio plays music having a mood similar to the one provided by the user in input.

6 <http://www.last.fm>

7 <http://musicoverly.com/>

8 <http://musicoverly.com/aboutus/aboutus.html>

1.2.2.4 The Echo Nest

The Echo nest⁹ is not a real music recommendation application but a music platform which can be used through a set of APIs to develop music recommendation applications. Among the other features, the APIs provide, giving a song as input, information about other songs similar to the first. The similarities are computed combining a content-based approach and a cultural-context analysis. In particular, the system analyzes a song extracting music features from the audio, using a series of psychoacoustic filters that imitate the human hearing mechanism. Furthermore, the system finds cultural relations between songs through a semantic analysis of related web pages, for example analyzing the terms used to describe a song. The cultural analysis works as a pre-filter to the acoustic engine: the acoustic analysis is in fact computed just on songs belonging to the same cultural cluster¹⁰.

⁹ <http://the.echonest.com/>

¹⁰ https://echonest-corp.s3.amazonaws.com/docs/whitepapers/Song2Song-1_0.pdf

2

Motivations, goals and state of the art

2.1 Overspecialisation problem

Music recommender systems have reached in the last years a good level of accuracy; in particular collaborative filtering recommender systems such as Last.fm have succeeded in exploiting the huge number of subscribers, providing (after a training period) recommendations in line with the users' interests, based on the ratings (explicit or implicit) of other like-minded users [Adomavicius and Tuzhilin, 2005].

Music recommender systems, however, still fail in discovering users' latent interests: they often suggest songs that, although accurately tailored on the users' past behaviour, do not take enough into consideration how the users' taste can evolve in the future. When a user profile is very focused in terms of content experience, the user is recommended with suggested songs that he likes, but to which he would be anyway exposed through other channels. For example a person that usually listens to hard-rock music only can have a latent passion for electro music, without knowing this, just because the social network (both real and virtual) he is part of is not much exposed to this kind of experience and the typical recommender system does not even try to propose it.

For this reason we argue that recommender systems should contaminate users experience with dissimilarity: if an appropriate level of accuracy is maintained, dissimilarity could increase users' satisfaction and stimulate latent interests.

2.2 Diversity, novelty, serendipity and user satisfaction

It is important to state the difference between diversity, novelty and serendipity in the context of recommender systems. The term diversity can have several meanings, but probably in the most common one refers to how diverse are the recommendations proposed by a recommender system: for example if a movie recommender system proposes just one genre of movies (e.g. “action movies”) the diversity of the suggestions provided is very low, regardless to how novel are for the user.

Novelty refers to how novel is a recommendation respect to what the user already knows, the “non-obviousness” of the recommendation; if the user already knows the items proposed (for example he already knows the movie proposed) the recommendation is not novel. Serendipity, following the definition of [Herlocker et al., 2004], should add another element: a serendipitous recommendation helps the user to find a surprisingly interesting item he might not have otherwise discovered. [Herlocker et al., 2004] provided a clear and meaningful definition of serendipity, which is, however, not very easy to apply when the task is to recognize, in a real-world recommender system, which recommendations are serendipitous.

Let's explain the difference between novelty and serendipity with an example: if a fan of the director Woody Allen receives a recommendation about a new Woody Allen movie, and he does not know it before; the recommendation is novel, but the user would probably have discovered the movie by himself. Let's consider a second example: a recommender system suggests an independent drama movie to a user who mainly watches mainstream action movies, the user, surprisingly, enjoys the movie; this

would be a serendipitous suggestion: probably the user would not have been exposed to that movie without the help of the recommendation.

There is therefore a main difference between novelty and serendipity and it is about the ability of recommender systems to propose non-obvious, surprising, suggestions; another implicit difference is about the success of the recommendation: in literature the concept of serendipity is in fact usually associated to a surprising *and* interesting suggestion; in other words, not only the suggestion should be apparently far from the current user's taste, but the user should also like it; the concept of novelty, instead, does not usually include this last assumption (likeness). Table 1 shows a tabular representation of the novelty and serendipity produced by six different combinations of appreciation and expectedness.

	Liked items	Disliked items
Known items	//	//
Unknown but expected items	Novelty	Novelty
Unexpected items	Novelty and Serendipity	Novelty

Table 1: Novelty vs. Serendipity

If we consider the concepts of diversity and serendipity, we can now state that, if a serendipitous approach is likely to also provide some diversity; a recommender system that just promotes diversity does not necessary provide serendipity: back to the movies recommender system example, we can in fact imagine that an eclectic user can be familiar with several kinds of movie genres and get suggestions that, although diverse, are not serendipitous.

There is an evident trade-off between accuracy and serendipity: a recommender system pursuing serendipity has necessary to take the risk to propose some bad recommendations in order to promote unfamiliar items.

Only concentrating on accuracy, however, can negatively impact the system itself as stated in [McNee et al., 2006].

Previous work in the field of political news recommendation [Munson and Resnick, 2010] found that there are some people who are diversity-seeking while others are challenge-averse; for diversity-seeking people, it is important to read not only comments and news which reinforce their existing viewpoints but also content coming from a different viewpoint: their satisfaction in fact, after a certain amount of agreeable news got, decreases. It is reasonable to think that a similar phenomena also occurs in the music field, where some eclectic people enjoy discovering music not exactly customized on their current taste while other people prefer conservative suggestions. It is also important to highlight that new users can have different needs from experienced users in a recommender system: while at a early stage conservative suggestions can be useful in order to make the system trustable, experienced users expect more serendipitous suggestions [Sean et al., 2006].

Not only the discovery attitude can change from user to user and it is usually lower in the first period of use, we also argued, as anticipated by [Swearingen & Sinha, 2001] that the same user can, anytime, experience two different fruition moments: one during which the user prefers to receive conservative suggestions, and a second one during which the user is willing to accept more adventurous suggestions. In order to have some further evidence about this aspect, we also published a Web survey asking how a music recommender radio should act; the survey contained just one question and an introduction to the topic. In figure 4 a screenshot of the survey is presented.

Users could optionally add a textual comments. The survey was published on the Web through the survey service [surveymonkey](http://surveymonkey.com)¹¹, advertised through the Facebook page of the author of this work and

¹¹ <http://surveymonkey.com>

publicly open to anyone who wanted to participate without requiring any information or login. Although it cannot be considered a formal scientific evidence it gives some clues about our guess; 39 users answered the survey and a screenshot of the results is reported in figure 5.

2 - Motivations, goals and state of the art

Question about music recommender radios

Uscire dal sondaggio

A suggestion by a music recommender radio can be:

- CONSERVATIVE if it is similar to something you already like, for example if you like THE BEATLES a very CONSERVATIVE suggestion could be JOHN LENNON, a less but still CONSERVATIVE suggestion could be THE ROLLING STONES;
- SURPRISING if apparently it's not directly related to what you already like but the system thinks you can still like it

A CONSERVATIVE approach has typically more chances to give CORRECT (liked) suggestions; on the other side, a SURPRISING approach has typically more chances to let the user DISCOVER new music lands he didn't know before.

1. A music recommender radio should

- Let the user choose when he/she wants conservative or surprising suggestions
- Propose music using a mix of conservative and surprising suggestions
- Propose music using just surprising suggestions
- Propose music using just conservative suggestions

Comments

Done

Figure 4: Screenshot of the Survey

1. A music recommender radio should		Create Chart	Download
		Response Percent	Response Count
Propose music using just conservative suggestions		0.0%	0
Propose music using just surprising suggestions	<div style="width: 2.6%;"></div>	2.6%	1
Propose music using a mix of conservative and surprising suggestions	<div style="width: 43.6%;"></div>	43.6%	17
Let the user choose when he/she wants conservative or surprising suggestions	<div style="width: 53.8%;"></div>	53.8%	21
	Comments		4
	Show Responses		
	answered question		39
	skipped question		0

Figure 5: Screenshot of the survey's results

The majority of the people (21 out of 39) preferred a recommendation radio which let the user choose each time between conservative and surprising approach; 17 people preferred a conservative/surprising mix, 1 person preferred a system which proposes just surprising suggestions; it is interesting to highlight that nobody expressed a preference for a recommendation radio which proposes just conservative suggestions.

2.3 Problem statement

The overall goal of this thesis is to propose techniques which can be adopted by recommender system designers in order to increase serendipity, in order to stimulate latent interests and, possibly, help users discovering music apparently far from their usually taste but still valuable. While simple random suggestions can increase the surprise factor, relying just on randomness can significantly decrease the accuracy of the suggestions and therefore the user satisfaction; an approach which increases the unexpected suggestion trying to preserve some accuracy is therefore needed.

The main goal stated above can be decomposed into six different subproblems; more in details, given a set A composed by n artists, a set U composed by m users and a set L of preference indicators, which represents (directly or indirectly¹²), for each user, the artists he likes, the six subproblems can be described as in the follows:

1. Propose a method for classifying the artists in different musical clusters, according to the aggregated taste of the users in the system, and to synthetically describe the content of each cluster.
2. Propose a method for measuring the eclecticism level of each user, according to the musical clusters he likes more.
3. Propose a method for selecting which musical clusters are more likely to contain serendipitous music for the user.

¹² More details on how the preferences can be derived are presented in chapter 3

4. Propose a serendipity metric, adoptable in the context of a real music recommendation radio.
5. Propose a method for computing a playlist which exposes the user to music which is, gradually, closer to the serendipitous musical cluster.
6. Develop a recommendation radio which implements the techniques proposed in order to evaluate in a real-world scenario some of the results.

2.4 Related work

The related work is presented according to each of the first five subproblems described in chapter 2.3 .

About the first one, automatic artists or songs classification is an important problem in the music information retrieval field; typically, the input data for the clustering process consists of extracted audio features or metadata (e.g. access logs, keywords co-occurrences on Web pages, ...). In literature we can find lots of work in automated methods for mapping artists or songs to a genre.

In [Knees et al., 2004] The authors used Support Vector Machine as the main approach to describe and classify artists according to word occurrences on related Web pages, having as a goal to organize artists by genre.

In [Shavitt and Weinsberg, 2009] the k-means clustering algorithm was used to cluster over 500,000 songs using peer-to-peer co-occurrences.

In [Tacchini and Damiani, 2011] the k-means algorithm was used to cluster about 20,000 songs belonging to five pre-chosen genres, using as input data some access logs related to the Last.fm music service.

In [Wang and Ogihar, 2010] about 400 artists were clustered using several clustering methods including K-means, Spectral clustering (Ncuts) and Non-negative matrix factorization and applying both a tag approach and

a content analysis approach. The goal was to to classify the artists on the base of their style.

Regarding the second subproblem, some preliminary results are shown in one of our previous work [Tacchini and Damiani, 2011], where we applied a first attempt in trying to measure the eclecticism level of a recommender system users and we also presented some results that seemed to highlight there was a correlation between the most liked cluster and the eclecticism level; in the experiment conducted, for example, Jazz lovers appeared more eclectic than Heavy Metal lovers.

About the third problem, to the best of our knowledge, the main literature is the following.

[Toms, 1999] was probably one of the first work who highlighted the importance of serendipity in information filtering.

In [Abassi et al., 2009] the authors propose the concept of “out of the box” (OTB) recommendation; they identify “regions” of movies the users are underexposed to and propose an algorithm which rates each item according to both accuracy and OTB-ness in order to try to favour items coming from regions the user is less familiar with.

While [Hijikata et al., 2009] is a work on novelty and not on serendipity, the authors propose an interesting approach to guess items that a user does not know: after having collected profiles of acquaintance by letting users give ratings of acquaintance for a set of items, they apply a collaborative filtering algorithm to profiles of acquaintance, identifying, for the target user, the group of users that is similar to him (from the acquaintance point of view). Then, the idea is that the items unknown to the group are considered unknown to the target user as well.

In [Iaquinta et al., 2009] the authors proposed a method to identify and suggest novel items in the context of a content-based recommender systems for cultural heritage fruition, and to properly arrange the items in a physical space.

In [Onuma et al., 2009] the collaborative filtering space is represented as a bipartite graph where a user (element of the users set of the graph) is connected to an item (element of the items set of the graph) if he likes the item; the authors propose an algorithm aimed at proposing “border” items i.e. items that, although not the most relevant for a user, are well connected with other group of items, and can serve to broaden the user's horizons. The main drawback of this work is that the experimental part is not complete: while the authors show results about the amount of “surprising” items suggested by the algorithm proposed, higher than the one provided by a normal approach, there is no evidence about the accuracy of the suggestions, i.e. how the users liked the surprising suggestions.

Finally, there is a lot of literature about diversity. As explained in chapter 2.2, the term diversity in the recommender systems field usually refers to the so-called *intra-list diversity* i.e. how much the items recommended are diverse. While diversity can stimulate serendipity, if the items proposed are similar to the ones the user already know, the level of serendipity does not increase. Two of the works about this topic are [Hurley, N. and Zhang, M., 2011] and [W. Zeng, W. et al., 2010].

Another work that is worth to mention is [Slaney, M. and White, W., 2010], because the term diversity in this case is closer to the concept of serendipity; while it is focused on content analysis, it provides an interesting approach: the authors of the patent suggests to represent the diversity of a user's taste in a multidimensional space, according to the dimensions provided by the content features, and to use a geometric figure which fits the representation as a the main input source for the recommendation. Each item that, in the same multidimensional space, can be represented inside the borders of the geometrical figure can be in fact safely recommended to the user. A less conservative approach consists in increasing the volume of the figure in order to propose to the user less obvious recommendations, adjusting the volume's change according to the user feedback.

We conclude this part of our literature review mentioning a very recent work, presented in late October 2011, [Oku, K. and Hattori, F. 2011], based on the following idea: items which have mixed features of two items the user likes are potentially serendipitous items.

About serendipity metrics, just a few attempts have been made. In [Murakami et al., 2008] the idea behind the proposed metric is that unexpectedness is low for easy to predict items and high for difficult to predict items; the authors, to estimate how an item is easy to predict, introduced the concept of primitive prediction method (PPM), supposing that unexpectedness is low for items that a PPM can predict and high for items that a PPM cannot predict, and finally considering unexpectedness, in a top-N recommendation problem, as the deviation from the result provided by a PPM. More in details, the two metrics proposed are the following ones: *unexpectedness* and *unexpectedness_r*, where the latter also takes into consideration the position of the item in the top-N recommendation list.

$$unexp. = \frac{1}{N} \sum_{i=1}^N \max(Pr(s_i) - Prim(s_i), 0) \cdot isrel(s_i) \quad (2)$$

$$unexp.r = \frac{1}{N} \sum_{i=1}^N \max(Pr(s_i) - Prim(s_i), 0) \cdot isrel(s_i) \cdot \frac{count(i)}{i} \quad (3)$$

In both the equations, s_i ($i = 1 \dots N$) denotes the i -th ranked item in the top-N recommendation list, $Pr(s_i)$ denotes the degree to which the recommender system confidently recommends the item s_i , $Prim(s_i)$ the degree to which a PPM confidently recommends the item s_i . and $isrel(s_i) \in \{0, 1\}$ denotes if the item is related to user's preferences (1) or not (0).

In [Ge et al., 2010], the metric introduced by [Murakami et al., 2008] was extended by taking into consideration only the suggested items that are

actually useful to the user. Considering RS as the set of recommendations generated by the recommender system and PM as the set of recommendation generated by a PPM, they defined the set of unexpected recommendation as:

$$UNEXP = RS \setminus PM \quad (4)$$

and the serendipity of the set as:

$$SRDP = \frac{\sum_{i=1}^N u(RS_i)}{N} \quad (5)$$

where $u(RS_i) = 1$ when the recommendation is useful to the user and 0 otherwise, RS_i is an element of $UNEXP$ and N the number of items in $UNEXP$.

While we can safely assume that a high level of serendipity should imply a high value of the two metrics proposed (*unexpectedness* and *SRDP*), it is also safe to state that the opposite is not always true: a recommendation that cannot be predicted by a PPM can still be expected by the user; furthermore it seems there is not agreement about which PPM to used.

In a [Oku, K. and Hattori, F. 2011] a modified version of the measures introduced by [Ge et al., 2010] is proposed: *r-unexpectedness* and *r-serendipity*, represented by the number of unexpected (*r-unexpectedness*) and serendipitous (*r-serendipity*) items over the number of items suggested by a top-N recommendation algorithm.

In [Zhou et al., 2010], the unexpectedness of a recommendation is measured using the *self-information* of an object. This indicator is known in information theory, however, since the self-information of an item is relative to its global popularity, we think it is not suitable to measure serendipity,

which is, instead, highly subjective. For example a very niche item i_1 could still be expected for a user whose interests includes items very similar to i_1 .

We think that all the measures proposed does not take enough into consideration one important factor: the cost the final user has to pay to get serendipitous recommendations. Imagine for example a music recommendation scenario: how many bad recommendations a user has to receive (i.e. how many bad, disliked songs a user has to listen to) in order to get a serendipitous one? We think this is a very important element to take into consideration: if the cost of receiving serendipitous recommendations is to high the user satisfaction could decrease until the user could eventually stop using the system. Only a very recent work [Oku, K. and Hattori, F. 2011] seems to partially take this into consideration, but still the bad (disliked) recommendations are not considered as a separate element of the *r-serendipity* equation.

About the last subproblem (number 5) there is a lot of work that focus, in general, on the playlist generation problem. [Fields, 2011] and [Balkema, 2009] are two Ph.D. thesis where a review of the literature about playlist generation is presented.

The work that approaches closer our problem is [Flexer et al., 2008]; the authors proposed a method to produce a playlist of N songs given the first and the last one, providing a sequence of songs in which the transition between one song and the following one is smooth, i.e. the songs at the beginning of the playlist sound like the first one, the songs at the end of the playlist sound like the last one and the songs in the middle sound like both the firsts and the lasts. The approach is based on audio similarity and after having computed the ideal position (in term of distance from the first and the last song) of each hypothetical intermediate song, it finds the song which is the closer to such ideal positions. The authors stated that, while they using content-based similarities, other kind of similarity measures can be used.

This work has an implicit assumption that is actually not verified when we have to deal with non-metric spaces like the one we are working with; let's introduce the concept with an example. Let's consider the distance between two songs as $(1-PCs)$, where PCs is the Pearson Correlation similarity between the songs (see chapter 3.2.3); if the distance between the song A and the song B is x , the sum of the distance between song A and song C (where $C \neq A$ and $C \neq B$) plus the distance between songs C and song B can be $< x$; in other words, the triangle inequality is not satisfied.

If the triangle inequality is not satisfied, considering the position of each intermediate song in the playlist only as distance from the first song and the last song, can lead to problems: for example, if we want to create a 4-songs playlist having just the first and the last available, choosing the song number 3 and number 4 only considering their distances from songs number 1 and number 4 does not give us any guarantees about the distance between song 2 and song 3. It is therefore not possible to set constraints which always guarantee a smooth transition between a song and the following one, providing a limit to the distance between them.

[Knees, 2006] uses a graph search approach similar to the one we present in chapter 3.6 to order the songs in the playlist once the composition of the playlist is decided; this approach, however, is focused on solving the problem of generating an ordered playlist given the songs but cannot be applied to our problem: in our case we only know the first and the last song and, as we present in detail in chapter 3.6.3 , we want to compute a playlist that gradually expose the user to the kind of music represented by the last song.

3

Proposed approach

Note: part of the paragraphs 3.2 and 3.3 is extracted from our previous works:

- Tacchini, E. and Damiani, E. 2011. What is a musical world? An affinity propagation approach. In Proc. of the 1st ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies.
- Tacchini, E. 2008. Toward trust-aware folksonomy evolution for blogosphere learning. Presented at Advanced international summer school: managing uncertainty and competition through dynamic learning networks; Ostuni, Italy.

3.1 Dataset

The dataset used for part of this research originally contained listening data of approximately 360,000 unique users of the social music service Last.fm¹³. To better understand the content of the dataset, the Last.fm music service is briefly described in the following section.

Last.fm is a music Web service founded in 2002, acquired by CBS Interactive in 2007 and counting millions of active users. The service creates

¹³ <http://www.last.fm>

a music profile of each user according to the songs the users listen to. Those songs data come from three different sources:

- the user's music software players (e.g. Apple iTunes or Windows Media Player), through a software the user has to install on his personal computer, which sends the data to Last.fm during the listening;
- the user's music hardware players (e.g. Apple iPod/iPhone or Microsoft Zune), through a software the user has to install on his computer, which send the data to Last.fm when the user connect the device to his personal computer
- the Last.fm Web radio.

The Last.fm Web radio provides several modalities, including:

- tag station;
- artist station;
- user station;
- recommendations station.

A tag station provides music described using a particular keyword (“tag”) by the community, where the chosen tag is provided by the user; an example could be the “Hard rock” station. A tag is not, however, always related to a music genre: for further details about the use of tags in music services see paragraph 3.2.4 . An artist station provides music by a selected artist (chosen by the user) and other similar artists (similar according to the Last.fm algorithm). A user station provides music by artists who are in the library of a user (coming from one of the three sources discussed above): a user can play his own user station but also station by some other user. The recommendations station play music by artists that the recommendation engine of Last.fm considers an interesting suggestion for the user.

The process of recording the information about the listening of a song is called *scrobbling*, a *scrobble* is recorded only if the user listen to at least the 50% (by default) of the duration of a song and users can also disable the scrobbling. Last.fm provides a set of APIs¹⁴ that can be used to retrieve information about scrobbed tracks. Oscar Celma, using those APIs and in particular the `user.getTopArtists()` method¹⁵ built a dataset containing about 15.7 million <user, artist, plays> tuples collected during Fall 2008; in particular, for each user, the dataset provides playcount information for a limited number of the user most played artists. The dataset is available for download for non commercial use¹⁶.

The dataset contains both music that users voluntary played and music coming from a recommendation radio; while it would be nice, from the research point of view, to extract just the first category of data, this is not possible because Last.fm does not provide this information. The dataset, however, can be still considered a valid resource for research in the music recommendation, especially considering that some of the radio mode (e.g. the tag station) strongly depends on the input of the users and also that if the user skips a recommended track before having played the 50% (default) of the track itself, the track is not *scrobbed*, and therefore it is not considered played. Furthermore, considering the scrobbles coming from the radio stations, allows to take into consideration artists that the user like but that he would not have played on his music player, for example because he does not have the audio files.

To better understand how precise is the data, it is necessary to explain more in detail how the data itself is collected. While the name of the artist and the title of the song should be always correct if a song is played by the Last.fm radio, they can be less accurate when a song is played by the user's

14 <http://www.last.fm/api>

15 <http://www.last.fm/api/show?service=300>

16 <http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-360K.html>

personal software/hardware player, in this case those information can be derived from two different sources:

- Track's fingerprint
- ID3 tags

Fingerprinting is a technique used for producing a unique ID for each song according to the actual audio content of the track; this technique allows to identify properly each track and it's enabled by default in the Last.fm software; sometimes the fingerprint could have been not available, for example because a user disabled it, in these cases the information come from the ID3 tags. ID3 is a metadata container, usually combined with mp3 audio files, which allows to store in the file some information about the track such as the author, the title, the album or the year; since the ID3 tags can be modified by users, this metadata is not necessary correct, it can contain typographical or other kind of errors and therefore some scrobbles can be affected by this lack of accuracy.

The original dataset used contained, for most of the artists, also a MusicBrainz¹⁷ Identifier. MusicBrainz is a project aimed at creating an open music database, containing information about artists, albums, tracks and relation among them and where each artist is unique identified by a MBID, that is also provided, if available by the *user.getTopArtists()* method of the Last.fm APIs. Furthermore, for some of the artists the Last.fm APIs didn't provide an MBID, Celma applied some heuristics to try to assign them one.

For the purposes of this research, we considered only tuples which provides an artist MBID. This filter should provide a higher level of confidence respect to solely rely on artist names, which, as explained before, can contain mistakes and therefore refer to the same artist while having a different name. Furthermore, we chose a subset of the original dataset, consisting of the 3,000 most popular artists and the first 100,000 users. This

¹⁷ <http://musicbrainz.org/>

size of the dataset allowed for fast computation time. An alternative version of the dataset consisting of the 30,000 post popular artists have been also created; to avoid confusion, let's call dataset_3000 the first one and dataset_30000 the second one.

3.2 Similarity measures

3.2.1 Introduction

In order to solve our first and fifth subproblems, namely: artists' classification and serendipitous playlist generation (see chapter 2.3) and, in general, for every problems which required to measure how similar are two artists, we needed a similarity metric. We decided to use an approach based on pure items-based collaborative-filtering, where the rating given to an artist by each user is implicitly deducted from the number of times a user listened an artist in the past.

The reasons why we chose a collaborative-filtering approach are mainly two:

1. Collaborative filtering has been used in previous work to find similarities between music artists and, within the Music Information Retrieval community, studies have shown that collaborative filtering systems consistently outperform content-based methods when applied to this task, at least for popular artists [Barrington et al., 2009];
2. It is easier (respect to a content-based approach) to adapt for using in other domains than music.

The two collaborative-filtering-based similarity measures used are described in chapters 3.2.2 and 3.2.3 : a simple boolean-similarity approach and an approach based on Pearson Correlation.

The datasets on which both the approaches have been applied derives from the one described in chapter 3.1 : dataset_3000 and dataset_30000; in particular the tuples from the Last.fm dataset were aggregated in order to create a matrix $M(3000,100000)$ (or $M(30000,100000)$ for dataset_30000) where each row i corresponds to an artist, each column j to a user and each cell $M(i,j)$ to the number of times user j played a song by artist i . We call this matrix play-counters matrix and in figure 6 an example is provided.

	User1	User2	User3	User4	User5
Beatles	45	10	0	10	50
Radiohead	20	50	12	0	5
Metallica	25	30	0	0	0
Miles Davis	0	12	0	20	120
Madonna	5	0	0	0	0
The Prodigy	0	0	30	0	0
Daft Punk	0	0	50	40	0

Figure 6: An example of play-counters matrix

In particular, since the affinity propagation clustering technique we used for classification (see paragraph 3.3.2.1 for details) requires pairwise similarities as input, we computed a symmetric artists-artists matrix $S(3000,3000)$ where each cell $S(i,j)$ contains a value representing the similarity between artist i and artist j .

A second group of similarity measures, folksonomy-based, is described in chapter 3.2.4 ; as we will explain in details later, we didn't directly use this second group of similarity measures to achieve our research aims; instead, we use them to validate some of the techniques proposed.

It is important to highlight that not only for artists classification but even when we needed to measure the similarity between two songs (we need it for playlist creations, see chapter 3.6) and, in general, everywhere in this thesis, we worked on an artist-level (evaluating the artist who plays the

songs) and not on a song-level, assuming that similarities between artists can approximate similarities between songs. While this could be considered a simplification (some artists explore several music genres, styles, topics, ...) artists similarity has been used as indicator of their songs similarity in the past, in literature (see [Knees et al., 2006] for an example) and in commercial application (see [Barrington et al., 2009] for an informal experiments that shows how probably Apple iTunes Genius¹⁸ works on a artist-level and not on a song-level). The reasons are essentially two:

1. Considering items instead of songs increase drastically the number of items and consequently the computation time of a typical collaborative-filtering problem;
2. The collaborative-filtering data about a song (in term of number of users who played it and number of time) is of course not as rich as the data about an artist and for some songs could be not rich enough to give reasonable recommendations.

Our third subproblem (selecting which musical clusters are more likely to contain serendipitous music for the user) also needs similarity metrics, but in this case to measure how similar are two users; specularly, we adopted an approach based on a variation of a pure users-based collaborative-filtering; all the details about this approach will be explained in chapter 3.5 .

3.2.2 Boolean similarity

To apply the boolean approach, we converted the matrix M into a matrix B where:

$$\begin{aligned} B(i, j) &= 1 \text{ if } M(i, j) > k \\ B(i, j) &= 0 \text{ otherwise} \end{aligned} \tag{6}$$

¹⁸ <http://www.apple.com/itunes/features/#genius>

This method convert the play-counters matrix M into a Binary “like/do not like” matrix, assuming that:

1. if a user listens to an artist more than k times, he likes the artist;
2. high play-counter values (i.e., $M(i,j) \gg k$) correspond to uniformly high liking, regardless of their values.

The threshold parameter k was set to 10 for the most popular artists and to 5 for all the others. Setting k to 0 would have brought to an oversimplification of the problem: a user can listen to a songs by an artist even if the artist does not match his musical taste, for example just because he has heard of the artist and want to try; furthermore, popular artist are more likely to get played without a relation with the user’s tastes so k should change according to the artist popularity.

As a popularity measure for each artist i we used the number of users $p(i)$ who listened at least once artist i . Defining “popular” artists those having $p(i) > plim$, where $plim$ is a dataset-dependent parameter we chose ($plim = 3,300$), we identified 172 “popular” artists so we set $k=10$ for these artists and $k=5$ for the remaining 2,828 ones.

We computed the similarity between each pair i,j of artists using the Jaccard index, as follows:

$$S(i, j) = \frac{|users(i) \cap users(j)|}{|users(i) \cup users(j)|} \quad (7)$$

where $users(i)$ is the set of the users who likes artist i . The result is a float number in the range $[0,1]$; 1 is the maximum similarity level.

This straightforward approach gives quite good results. For example, top five similar artists to “The Chemical Brothers” computed this way were other electronic-music artists: “Fatboy Slim”, “Daft Punk”, “The Prodigy”, ”Moby” and ”Massive attack” (sim.: 0.158, 0.157, 0.156, 0.130 and 0.130).

3.2.3 Pearson Correlation similarity

For the Pearson Correlation approach we considered the play-counters in M as the extent to which users like artists, assuming that higher play-counter values means higher levels of liking. In this case a normalization of play-counter values is needed to avoid distortions due to the difference in popularity between artists; we therefore computed a normalized matrix N by multiplying each row of M by a factor s that makes the sum of the squares of the row equal to one. Then, we measured the similarity $S(i,j)$ between each pair i,j of artists as follows:

$$S(i, j) = \frac{\sum_{x=1}^u ((I_x - \bar{I}) \times (J_x - \bar{J}))}{\sqrt{\sum_{x=1}^u (I_x - \bar{I})^2 \times \sum_{x=1}^u (J_x - \bar{J})^2}} \quad (8)$$

where I and J are two rows of the matrix N , corresponding to artists i and j ; while \bar{I} and \bar{J} are the means of I and J ; I_x is the x^{th} element of the row I ; J_x is the x^{th} element of the row J and u is the number of users considered (100,000 in our case).

According to the similarities computed this way, the top five similar artists to “The Chemical Brothers” were: “Fatboy slim”, “The Crystal Method”, ”The Prodigy”, “Daft Punk” and “Underworld”. These results are quite similar to the ones we got using the boolean approach. To give a first general idea about how this approach and the boolean work, in the following tables we present, for “The Chemical Brothers” and other five popular artists (Metallica, Bob Dylan, Madonna, Jay-z and Ramones), the top 10 similar artists provided by both the approaches and the top 5 similar artists provided by the Pandora Radio¹⁹. The intersection between the artists

¹⁹ <http://www.pandora.com>, in the details page of each artists a list of 5 similar artists is provided

3 - Proposed approach

computed by the approaches proposed and the ones provided by Pandora are highlighted in grey. While we do not consider Pandora as a ground truth for computation of artists similarities, the comparison is interesting because the results of Pandora come from the analysis done by music experts.

Boolean	Pearson Correlation	Pandora
Iron maiden	Iron maiden	AC/DC
System of a down	Megadeath	Ozzy Osbourne
Guns 'n' roses	Pantera	Black Sabbath
AC/DC	Slayer	Alice In Chains
Rammstein	AC/DC	Disturbed
Megadeath	System of a down	
In flames	Guns 'n' roses	
Red hot chili peppers	Rammstein	
Nightwish	Black Sabbath	
Slayer	Ozzy Osbourne	

Table 2: Artists similar to Metallica

Boolean	Pearson Correlation	Pandora
Fatboy slim	Fatboy slim	The crystal method
Daft punk	The crystal method	Daft punk
The prodigy	The prodigy	Fatboy slim
Moby	Daft punk	The prodigy
Massive attack	Underworld	Dadmau5
Groove armada	Orbital	
Röyksopp	Massive Attack	
Faithless	Basement Jaxx	
Underworld	Groove armada	
The crystal method	Moby	

Table 3: Artists similar to The Chemical Brothers

3 - Proposed approach

Boolean	Pearson Correlation	Pandora
The rolling stones	Neil Young	The band
The Beatles	The rolling stones	Neil Young
Neil Young	The Beatles	The Beatles
Johnny Cash	The band	Buffalo springfield
Daive Bowie	Leonard Cohen	John Lennon
Tom Waits	Tom Waits	
Led Zeppelin	The who	
Katy Perry	Grateful dead	
Bruce Springsteen	The kinks	
Pink Floyd	Johnny Cash	

Table 4: Artists similar to Bob Dylan

Boolean	Pearson Correlation	Pandora
Britney Spears	Kylie Minougue	Cyndi Lauper
Kylie Minougue	Britney Spears	Michael Jackson
Rihanna	Barbra Streisand	Eurythmics
Nelly Furtado	Beyoncé	Lady Gaga
Beyoncé	Janet Jackson	Kylie Minogue
Michael Jackson	Cher	
Christina Aguilera	Rihanna	
Lady Gaga	Gwen Stefani	
Justin Timberlake	Christina Aguilera	
Mariah Carey	Mariah Carey	

Table 5: Artists similar to Madonna

3 - Proposed approach

Boolean	Pearson Correlation	Pandora
Nas	Nas	Notorious B.I.G.
Kanye West	Notorius B.I.G.	Kanye West
Notorius B.I.G.	Kanye West	Jaz-z & t.i.
Common	Common	Eminem
Lil Wayne	Lupe Fiasco	Nas
The Game	Camron	
Lupe Fiasco	Young Jeezy	
t.i.	Ice Cube	
50 cents	az	
Snoop dogg	t.i.	

Table 6: Artists similar to Jazy-z

Boolean	Pearson Correlation	Pandora
The clash	The clash	The clash
Misfits	Misfits	Joey Ramone
Rancid	Sex pistols	Sex pistols
Bad religion	Screeching weasel	Social distortion
No fx	Buzzcocks	The misfits
Sex pistols	Dead kennedys	
Dead kennedys	Rancid	
AC/DC	Bad religion	
The rolling stones	The exploited	
Johnny Cash	Pennywise	

Table 7: Artists similar to Ramones

3.2.4 Folksonomy-based similarities

3.2.4.1 Introduction to folksonomies

The origin of the word *folksonomy* lies in a combination of the words *folk* and *taxonomy* and refers to a classification system based on the process of annotating content using keywords by a community of users, not necessary experts in the domain. The keywords, called “tags”, can be freely chosen by the users and they are not part of a controlled vocabulary.

A very popular method to graphically represent the tags associated with an item is the *tag cloud*. In a tag cloud all (or the most used) tags that people used to describe an item are listed (usually alphabetically) and the importance of each tag (i.e. the number of users who used that tag) is represented by the font size: the bigger is the font size, the higher is the number of people who used it. This approach allows to easily understand which are the most important tags used, giving a quick idea about the content and also allows a faceted browsing of contents: each tag of the tag cloud is in fact often associated with an hyperlink, which allows the user to browse all the content items which have been described using the same tag.

Folksonomy has been used in several domains, for example pictures (see Flickr²⁰), bookmarks (see delicious²¹), books (see Amazon²²) and music (see Last.fm²³). In some domains or application where the content is produced by the user themselves, however, an item is usually tagged only (or almost only) by the content creator (and sometimes he is the only who has the right to do it); this could results in the creation of poor tag clouds.

3.2.4.2 Folksonomies in the music domain

In the music domain we can find a very interesting example of rich and useful folksonomy: the social radio services Last.fm provides a community-

20 <http://www.flickr.com/>

21 <http://www.delicious.com/>

22 <http://www.amazon.com/>

23 <http://www.last.fm/>

based descriptor for each artist, album and song: users can freely describe an artist or a song using one or more tags and a resulting tag cloud is created by aggregating all the tags. In figure 7 we present as an example the tag cloud of the band *Oasis*²⁴, where we can see that the most used term to describe this band is “britpop”, followed by “rock”, “british”, “alternative” and “indie”.

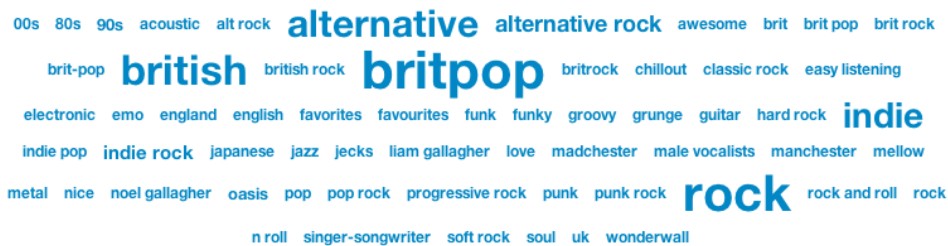


Figure 7: The Last.fm tag cloud for the band Oasis

While it is expected to find a rich tag cloud for a very popular band, since having more listeners means having more chance to be tagged, a surprisingly rich tag cloud can be found even for less popular band; as an example, in figure 8 we present the tag cloud of the band “Uochi Toki”²⁵, an Italian underground Hip-Hop band; to give an idea of the difference in audience, Oasis has 2,619,755 listeners on Last.fm²⁶ while Uochi Toki just 6,073²⁷. It is not very common to find this amount of folksonomy data in other domain and, even in the music domain, the Last.fm example is pretty unique; the reasons could be probably found in the characteristic of its community and in the easiness with which it is possible to tag an item (artist, album or song) using the “scrobble tool”²⁸, a software that users

24 <http://www.last.fm/music/Oasis/+tags/>

25 <http://www.last.fm/music/Uochi+Toki/+tags/>

26 <http://www.last.fm/music/Oasis>

27 <http://www.last.fm/music/Uochi+Toki>

28 <http://www.last.fm/download>

mainly use to listen to the Last.fm radio channels, “scrobble” songs and tag contents.



Figure 8: The Last.fm tag cloud for the band Uochi Toki

3.2.4.3 Music folksonomies and Music Information Retrieval / Recommendations

While it is evident the potential that folksonomies can give to help Music Information Retrieval or Music Recommendation problems, determining if the information provided can be trusted is a crucial issue. Since there is not a controlled vocabulary it is important to be aware of the following issues:

- As discussed in [Bischoff et al., 2008], folksonomy descriptors can include very different kinds of semantic information; while descriptors usually refer to music genre and style, sometimes they can also refer to different concepts, for example the mood (“melancholic”), a judge (“amazing”) or a period (“90s”);
- Folksonomies can include some noise tags, i.e. tags that do not really describe the content;
- Similar tags can express different semantic values.

About the last point, as stated in [Golder and Huberman, 2005] the semantic problems which can affect a folksonomy are:

- polysemy (same tag, different semantic value);

- synonymy (different tag, same or close related semantic value);
- basic level variation.

“Basic level variation” refers to the fact that different people may use words at different levels of specificity to properly tag a content: for example a picture of a collie dog could be tagged (from general to specific) as “animal” “dog” or “collie”; the same issue can also affect music folksonomy; an artist could for example be tagged as “Rock”, “Indie Rock” or “Brit indie rock”: while all three tags could be correct, they refer to different levels of specificity.

Finally, we should consider that in Last.fm, users can add tags even without listening to the relative content; this means that theoretically there could be for example tags added to an artist by users who never actually listened to that artist.

We should consider that the adoption of personal tags should be evaluated positively since it directly reflects the vocabulary of users [Mathes, 2004]; for this reason we can even find an example of the adoption of the folksonomy approach in the ontology world [Hepp et al., 2006]; the authors proposed to open the process of ontology creation to all the ontology users: through a Wiki platform anybody could add elements to an ontology or modify existing ones.

More than one work [Halpin et al., 2007, Quintarelli, 2005] showed that folksonomies tend to follow a power law distribution, where many people agree on using a few popular tags, through a process of social consensus. This behaviour seems to be positively correlated to the number of users and to the duration of the process, therefore, if the number of active users is high enough, over time users tend to agree on a small subset of stable tags [Halpin et al., 2007].

In order to further help this natural process of uniforming, some techniques have also been proposed in literature and in commercial

application. A popular strategy used to avoid tags proliferation is the publication of all tags commonly used by other users for the same content item; this approach is followed for example by the social bookmarking application delicious or by the social radio Last.fm, whose tagging application tool interface shows to the user its previous adopted tags and the community most used tags for a specific song.

Some research work tried to handle folksonomies with a structured approach: in [Gruber, 2007] an ontology of folksonomy has been proposed for modelling the tagging activity. This ontology allows for example to describe that a user tagged an object, that two tags represent the same concept or that a tag is not related to an object (useful for handling spam tagging) and can be a common base for sharing tags between different applications and users and for apply automatic reasoning about them. Following the idea of the tag not only as a word but as an object the faviki²⁹ application allows users to add tags to a Web page using Wikipedia concept as keywords; this approach obviously forces users to agree on a set of well-established tags and gives a semantic value to tags, providing some information about its properties and the connections with other tags. Another solution has been the adoption of a domain name for tags; the Flickr on-line community platform supports machine tags, which use a special syntax to define some extra information. Specifically, each machine tag is composed of a triple: domain, predicate and value; for example a user could tag a photograph with `geo:locality="Milan"` to assert that the shot has been taken in the city of Milan. Machine tags at the moment do not provide any controlled vocabulary or reserved domain space [Flickr forum discussion, 2007]. A more structured approach is the MOAT (Meaning Of A Tag) project [Passant and Laublet, 2008]. MOAT provides a framework which enables a semantic enrichment of free chosen tags: a blogger, after having assigned a tag to a post, can link it to a Semantic Web URI (such as

²⁹ <http://www.faviki.com>

dbpedia or geonames URIs), defining a meaning for the tag. The MOAT architecture is composed by an ontology, a server and some third-party clients.

If we consider, regarding the Last.fm folksonomy:

- the amount of data provided;
- the semantic richness;
- the high number of users (30 million people in 2009 who used last.fm monthly according to a post from the company's blog³⁰), who leads to a natural tags standardization process;
- the life of the community (almost ten years);
- the tools used to make the tagging process as user friendly as possible and at the same time to uniform the tags

we should consider the Last.fm folksonomy as an important source of information to exploit in order to evaluate similarities and classify music content and as a possible ground truth to evaluate approaches that make use of other sources.

It is important to highlight that an existing work [Sordo et al., 2008], analyzed the agreement between expert-based music genres vocabularies and music folksonomies using Last.fm data and coming to the conclusion that, while on some genres experts and wisdom of crowds agree, there are some other genres on which they disagree. Nevertheless we should consider that there is in general no strong agreement on the ground truth (even because music perception is highly subjective) and that a rich tag cloud contains information about music genre and style but also additional and important information influenced by the cultural factors (e.g. the mood

³⁰ <http://blog.last.fm/2009/03/24/lastfm-radio-announcement>

associated with a song or the topic of the lyric), therefore [Sordo et al., 2008] does not affect the credit we give to this approach.

In this thesis, we will not directly use the folksonomy approach as a base to solve our tasks (e.g. artist's similarities computation, artist's classification); instead we used to validate the results of a more general approach based just on pure collaborative filtering techniques. While combining our approach with a folksonomy-based might be useful in the music domain, our goal was to propose a technique which can be more easily adapted to other domains, where a so rich folksonomy usually is not available: implicit or explicit ratings is probably the data that can be easily collected, regardless of the domain.

3.2.4.4 Similarity measures

We used the last.fm API `artist.getTopTags`³¹ to collect the tag cloud for each of the 3,000 artists in the `dataset_3000` dataset; the `getTopTags` API provides, for each tag in an artist tag cloud, a normalized weight representing the importance of that tag in the artist's tag cloud.

We adopted three different custom similarity indexes: *tagsim*, *wordsim* and *textsim*, all based on artist's tag clouds.

The following pseudocode can be used to compute a *tagsim* index value between the tag clouds of two artists x and y ; where V is a vector which represents all the different tags available in the two tag clouds, X_i is the weight of the tag V_i for artist x and Y_i for the artist y .

```
foreach V as Vi
  tag_similarity = min(Xi, Yi)
  max_tag_similarity = max(Xi, Yi)
  num = num + tag_similarity
  den = den + max_tag_similarity
end foreach
similarity = num/den*100
```

³¹ <http://www.last.fm/api/show?service=288>

since the data provided by Last.fm is already normalized, the pseudocode above produces a score (range 0...100) which represents the percentage of tags' weights shared between two artists. Only the ten most important tags have been used for each artists's tag cloud.

A problem that can affect this approach is that it considers all the tags equally distant from the semantic point of view: e.g. “Heavy metal” and “Metal” are considered completely different, just as “Heavy metal” and “Classic”; we therefore propose even a variation, *wordsim*.

Wordsim, instead of considering tags, considers tag words: e.g. the tag “heavy metal”, weight 50, is considered as two different tags, namely “heavy” and “metal”, weight value 50 for both. While with wordsim we lose part of the semantics provided by tags composed by more than one word, this approach can partially solve the issue related to tagsim explained above.

Finally the third approach, *textsim*, is based on string similarity: while it is probably not well suited for tag clouds comparison, it can be an additional and different indicator for the evaluation of our results.

To compute a textsim index value between two tag clouds, each of them is firstly converted into plain text, writing each tag for a number of time corresponding to the tag's weight and followed by a blank space. The resulting texts are then compared using the text similarities implementation of PHP³², getting a score in percentage which represents how much the tag clouds are similar.

³² <http://php.net/manual/en/function.similar-text.php>

3.3 Artists' classification

3.3.1 Introduction

Artists' classification is the first of our subproblems (see chapter 2.3); in order to classify artists we adopted an approach based on the Affinity Propagation clustering algorithm and on the similarity measures described in chapters 3.2.2 and 3.2.3 .

3.3.2 Affinity Propagation clustering and Musical Worlds detection

3.3.2.1 The Affinity Propagation algorithm

Affinity Propagation is a relatively recent clustering algorithm, whose details have been published for the first time in 2007, which works on the concept of “exemplar”: an exemplar is an item that is well representative of itself and some other items. It has been used for various tasks, for example for the analysis of network traffic, for the detection of interesting points in images or for the identification of common trading patterns [Frey and Dueck, 2007].

The Affinity Propagation algorithm takes as input similarity values among pairs of items and considers all items, simultaneously, as potential “exemplars”. By considering all data points as candidate exemplars, Affinity Propagation, respect to other classic techniques that rely on an initial random selection of items (for example the centroids selection in k-means), is able to avoid many of the poor solutions caused by unlucky initializations.

The algorithm, in particular, works exchanging valued messages among items until a good enough set of exemplars and clusters emerge. There are two kinds of message exchanged between data points, the “responsibility” messages and the “availability” messages. The first kind of messages are sent from a data point i to a candidate exemplar k : e.g. $r(i,k)$ represents how suitable point k is to serve as an exemplar for point i , taking into account

other points that could be exemplars as well. The “availability” messages $a(i,k)$ are sent from candidate exemplars to a data point, e.g. $a(i,k)$ represents how appropriate is for point i to choose k as exemplar, taking into account other points that support k as exemplar. The algorithm accepts in input a set of preferences which represent, for each point, the a-priori suitability to become an exemplar; if, as often happens, all the data points are equally suitable to become exemplars, the same preference value has to be set for all the points. [Frey and Dueck, 2007]

Given a set $I\{i_1, i_2, \dots, i_n\}$ of items, a similarity function between the items $s(x, y)$ where $x \in I$ and $y \in I$ and a set of preferences $P\{p_1, p_2, \dots, p_n\}$ the algorithm's main steps are described in the followings [Frey and Dueck, 2007].

At the beginning, the availabilities are initialized to zero:

$$a(i, k) = 0 \tag{9}$$

Then, the responsibilities are computed using the rule:

$$r(i, k) = s(i, k) - \max_{k' \text{ s.t. } k' \neq k} \{a(i, k) + s(i, k')\} \tag{10}$$

In the first iteration all the availabilities are set to zero so $r(i,k)$ is equals to the similarity between point i and point k minus the largest of the similarities between point i and other points (which are potentially candidate exemplars). In the following iterations, when the availabilities are updated (see equation 12), for items which have a low level of availability, it is more difficult to become exemplar. [Frey and Dueck, 2007]

For $k = i$ the responsibility $r(k,k)$ is set to p_i minus the largest of the similarities between point i and other points :

$$r(k, k) = p_i - \max_{k' \text{ s.t. } k' \neq k} \{s(k, k')\} \quad (11)$$

The availability $a(i, k)$ is set to $r(k, k)$ plus the sum of the positive responsibilities candidate exemplar k receives from other points; in particular, only the positive responsibilities are taken into consideration:

$$a(i, k) = \min\{0, r(k, k) + \sum_{i' \text{ s.t. } i' \notin \{i, k\}} \max\{0, r(i', k)\}\} \quad (12)$$

Responsibilities and availabilities are updated at each iteration and each time the exemplar of a point i can be calculated as the value of k that maximizes:

$$a(i, k) + r(i, k)$$

The procedure may be terminated:

- when the *max_it* number of iterations is reached;
- if changes in responsibilities and availabilities are below a threshold;
- if the responsibilities and availabilities do not change for n iterations.

Max_it and n are accepted as a input parameter by the algorithm. [Frey and Dueck, 2007]

When availabilities and responsibilities are updated, they are damped in order to avoid unwanted oscillations which can occur; the new (dumped) value is set to λ time its old value plus $(1 - \lambda)$ time its (real) new value, where $0 \leq \lambda \leq 1$. [Frey and Dueck, 2007]

The resulting number of clusters does not have to be specified in advance, it depends on the similarity between the items but it is also

influenced by the set of preferences P : the higher are the P items' values, the bigger will be the number of clusters. [Frey and Dueck, 2007]

Affinity propagation has been tested and compared with another exemplar-based clustering algorithm, k-centers in two experiments: the clustering of images of faces and the detection of genes; in both the experiments affinity propagation got better results than k-centers. [Frey and Dueck, 2007].

3.3.2.2 Application to our Artists' classification problem

To the best of our knowledge, Affinity Propagation has never been applied to solve a collaborative filtering music classification problem. It has been, however, successfully applied to several research fields and problems like biology [Dueck et al., 2008], image organization/summarization [Jia et al., 2008] and video clustering [Xie and Wu, 2008].

In this work we refer to a “Musical World” (MW) as a set of artists which can be considered homogeneous by some criteria; a MW often has a strict relationship with a music genre or style, but the characterization of a MW can also relate to other aspects like the geographic origin of the artists (e.g. a “German” MW), the topic of the lyrics (e.g. a “Christian” MW), the association with a culture or lifestyle (e.g. an “Alternative” MW) or the context in which the music is usually played (e.g. a “Soundtrack” MW).

Clustering artists according to their similarity is a useful preliminary process for serendipitous recommendations: when the goal of a recommender system is to propose non-obvious suggestions, the system can, after having classified artists in MWs, determine which MWs the user still have to explore. For an example in movie recommendations, see [Abassi et al., 2009]).

In order to use Affinity Propagation for our aims, we applied a Matlab implementation of the algorithm to the similarity matrix S computed from matrix B , assigning -2 as a preference value to all the artists in the dataset.

Assigning the same value to all the items means we do not have any a priori preference about the choice of a particular artist as exemplar. For additional information about how the choice of the preference parameter value can affect the clustering results, especially in term of number of clusters, refer to [Frey and Dueck, 2007]. Using such a preference value we found 36 clusters.

We then applied Affinity Propagation to the similarity matrix S computed from matrix N (Pearson Correlation similarity). We chose and assigned a preference value (-3.315) which produced the same number (36) of clusters obtained before; this made the results comparison easier.

Finally, we applied an implementation of the classic k-means clustering algorithm to N , setting $k=36$, using Pearson Correlation as a distance measure and repeating the clustering process ten times to improve results. We used dataset_3000 for all the three tests.

For a second experiment we applied Affinity Propagation to a much bigger dataset of artists, considering the first 30,000 artists ordered by popularity, dataset_30000.

Clustering 30,000 artists, considering that in our case the similarity value $s(i,j)$ is equals to $s(j,i)$ and that we do not need to compute the similarity of an artist with himself, needs to compute $(30,000 \times 30,000)/2 - 30,000 = 449,970,000$ similarities. This is very expensive from the computational point of view, therefore we decided to conduct the experiment computing just the boolean similarity and not the Pearson Correlation similarity.

The boolean similarity computation can take advantage of some interesting matrix calculation properties:

1. To easily get, for each couple of artists, the number of common users (intersection), you can multiply the matrix B_2 by its transpose B_2' :

$$I = B_2 \times B_2'$$

getting an intersection matrix I ($30,000 \times 30,000$) which contains, for each couple x and y , the number of elements that, for both the rows x and y of the matrix B_2 are set to 1.

2. To easily get, for each couple of artists, the number of users who likes one artist or the other (union), you can exploit the fact that for two sets A and B , you have:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

We exploited the above two properties to build a Matlab-optimized version of the boolean similarity computation algorithm, that take also into consideration the fact that Matlab has a special representation format for sparse matrix³³ and that some operation are faster using the Matlab sparse matrix representation while other operations are faster using a full representation. The complete algorithm is here presented:

```
load('B');
B_sparse = sparse(double(B));
clear('B');
save('B_sparse.mat', 'B_sparse', '-v7.3');
union_length = sum(B_sparse,2);
union_length = full(union_length);
clear('B_sparse');
tmp = repmat(union_length, 1, length(union_length))
+ repmat(union_length, length(union_length), 1);
clear('union_length');
S = zeros(30000,30000);
load('B_sparse');
```

³³ <http://www.mathworks.com/help/techdoc/math/f6-32006.html>

```
S = B_sparse * B_sparse' ;  
clear(' B_sparse' );  
S = full(S);  
S = S ./ (tmp-S)*1000;34
```

Using this algorithm, we computed all the similarities among 30,000 artists in less than two hours using a Macbook Pro Personal Computer, CPU: 2.53 GHz Intel Core 2 Duo, RAM: 8 GB 1067 MHz DDR3m running Mac OS X 10.6.8 and Matlab 7.12.0.635 (r2011a) 64 bit.

The time needed to run the Affinity Propagation algorithm strongly depends on the number of similarities the algorithm receives in input. In particular, the algorithm needs to execute a number of scalar computations equals to a constant times the number of input similarities, where the constant time is very often 1,000 (default maximum number of iterations *max_it*). Therefore we can argue that, since for the algorithm the similarity value $s(i,j)$ is not necessary equals to $s(j,i)$ its complexity is quadratic in the number n of items to cluster, resulting in $O(n^2)$; for $n = 30,000$ running Affinity Propagation on a standard modern personal computer becomes quite difficult.

While setting the similarities for all the points to cluster is the optimal input for the algorithm, Affinity Propagation does not need that and can also work using only a smaller number of relevant similarities. We chose therefore to reduce the complexity of the algorithm discarding some of the similarities, and in particular the similarities which lie under a threshold *min_sim_cluster*; this approach allows to get a sparse similarities matrix and to exploit an alternative implementation of the algorithm³⁵ that keeps advantage of sparseness, exchanging messages just between pairs of points when the similarity value is available. The approach is reasonable: if for

³⁴ We multiplied the similarities by 1,000 to increase the speed of some subsequent operations; this is, however, not needed for the similarities computation

³⁵ <http://www.psi.toronto.edu/affinitypropagation/software/apclusterSparse.m>

example we know that two items have a similarity value equals to $2/1000$, we know in advance that we do not want the two items being in the same cluster so exchanging messages between them does not add much value to the clustering quality. Setting $min_sim_cluster = 4$ allowed to have in input ~ 18.7 million of similarities instead of 900 million ($30,000 \times 30,000$), drastically reducing the running time and keeping an acceptable precision level.

3.3.2.3 How to represent a Musical World

We propose a technique to represent a MW based on folksonomy and on exemplar artists. It produces an extremely compact but still informative representation. The idea is to aggregate the tag clouds of each of the musicians in order to intuitively and graphically represent the content of the MW, producing an illustration we called “Musical World Tag Cloud” (MWTC); the name of the exemplar artist for that MW, as found by Affinity Propagation, is also shown; since that artist is supposed to synthesize the characteristics of all the musicians belonging to the same MW, it can give a further indication of the content of the MW.

A Musical World Tag Cloud $MWTC_i$, assuming A_i as the set of artists belonging to MW i , can be described as a tag cloud:

- having as tags all the distinct tags available in each artist's tag cloud, for all the artists in A_i
- having as weight, for each tag, the sum of the weights, for that tag, of all the artist's tag clouds, for all the artists in A_i .

To avoid noise we took into consideration just the most used tags (a limit of 11 tags, for each MWTC, have been set). We created a MWTC for each Musical World.

3.3.2.4 Examples and first discussion

In table 8 and table 9 the representative artists found by Affinity Propagation using the Boolean similarity (Boolean AP) and the Pearson Correlation similarities (PC AP) are presented, together with the size of each MW and the most important tag of the corresponding MWTC. K-means does not provide a direct indication of the representative item of each cluster.

In figure 10 some interesting MWTCs (graphically generated by IBM Word Cloud generator³⁶) found by PC AP are presented.

The fact that a single tag, having a size much bigger than the others, usually emerge in a MWTC, confirms the homogeneity of the MW; in particular it is possible to intuitively recognize some MWs characterized by a music genre: Classical, Pop, Punk, Jazz, Hip-Hop but also some other MWs characterized by other factors like the nationality: German.

The Radiohead's MW is quite interesting because, while the MWTC is not as homogeneous as others MWs from the point of view of a traditional genre/style classification, most of the artists are associated with the “alternative” (or perceived as alternative) culture.

Also the soundtrack MW provides useful insights: we can argue that there is a significant community of users that use to listen to music which is composed for movies/TV series soundtracks.

In Figure 9 some MWTCs found by Boolean AP are presented. It is interesting to highlight that the Boolean approach produced some additional small clusters (e.g. “Japanese”, “Turkish”) which seems to be not strictly related to the music genre/style; while in some cases such clusters may not be suitable to be effectively used by recommender systems applications, they still provide some valuable insight into the users’ musical worlds, providing elements useful to understand musical phenomena.

³⁶ <http://www.alphaworks.ibm.com/tech/wordcloud>

Three “indie” clusters have been discovered, whose representative artists were: Sufjan Stevens, Animal collective and The pigeon detectives; while it could appear as a classification mistake (for all, the tag “Indie” seems dominant respect to the others), each of the three MWs shows quite specific style characterizations (in part also suggested by the other, minor, tags); looking at the artists lists, in particular, we could argue that the first cluster seems to be characterized by a more structured and soft music style, with acoustic guitar as key instrument, the second one by a more experimental and less structured approach, with more electronic and synths, and third one, maybe the most characterized, by indie rock with some punk rock influence, with use of guitar distortion. From the emotional point of view, without the pretension of providing a scientific evidence, the classification suggests a link, for the first MW, with melancholic/introspective emotions, for the second one with a creative/explorative mood and for last one with energetic moods and emotions.

While the PC AP highlighted a Hip-Hop MW, represented by the artist “Nas”, the Boolean AP discovered two Hip-Hop clusters: a first group of artists represented by “The game” and another one represented by “Gang Starr”. The list of artists suggests that the first cluster mainly contains mainstream artists and is closer to the Pop MW, while the second cluster contains a more underground, niche group of artists probably preferred by Hip-Hop lovers and experts.

The list of all the artists, together with the most important tag, classified by musical world using Boolean AP is provided in Appendix B; the list for the three methods compared can be found online³⁷. Furthermore, at the same URL, we published the MWTCs for all three methods compared. All the data presented is related to dataset_3000.

These first discussion findings provide an initial grounding to the intuitive notion of “Musical World”. It seems that most of the clusters are

³⁷ <http://mentorfm.com/mirum11/>

3 - Proposed approach

semantically rich and their semantics can be also related to other factors besides genres, factors which express different facets of a music ecosystem and in particular of the social and cultural context where artists act. In chapter 5 , dedicated to the results, we will present in details all the results related to the classification tasks.

Representative Artist	Cluster's size	Main tag
Radiohead	7.43%	Indie
The beatles	8.93%	rock
Linkin park	4.77%	rock
Death cab for cutie	4.93%	indie
Arctic monkeys	3.23%	indie
Madonna	4.23%	pop
Iron maiden	3.17%	hard rock
Nightwish	3.13%	Gothic Metal
Rihanna	4.73%	pop
Boards of canada	3.47%	electronic
Sonic youth	3.50%	indie
Miles davis	3.57%	jazz
Opeth	2.23%	Progressive metal
Animal collective	3.30%	indie
Bonobo	3.57%	electronic
Nofx	3.13%	punk
Nas	4.23%	Hip-Hop
Hans zimmer	1.93%	Soundtrack
Die Ärzte	1.73%	german
Cut copy	2.23%	electronic
Armin van buuren	2.00%	trance
As i lay dying	2.30%	metalcore
Underoath	1.70%	post-hardcore
Ensiferum	1.27%	folk metal
Cannibal corpse	1.27%	death metal
Vnv nation	2.00%	industrial
All time low	1.57%	rock
Chico buarque	1.40%	brazilian
Darkthrone	1.33%	black metal
Pidżama porno	1.27%	polish
Andrés calamaro	1.57%	spanish
Håkan hellström	1.27%	swedish
Have heart	0.87%	hardcore
Sizzla	1.13%	reggae
Franz schubert	0.87%	Classical
Newsboys	0.73%	christian

Table 8: Representative artists, PC AP

3 - Proposed approach

Representative Artist	Cluster's size	Main tag
Radiohead	3.80%	Alternative
The rolling stones	6.80%	rock
Sufjan stevens	6.37%	indie
Rihanna	5.63%	pop
Animal collective	5.17%	indie
Judas priest	3.70%	hard rock
Breaking benjamin	3.70%	rock
Die Ärzte	2.40%	german
Rancid	3.83%	punk
Ensiferum	2.60%	Melodic Death Metal
Autechre	2.93%	electronic
The game	1.93%	Hip-Hop
Epica	1.87%	Gothic Metal
Simian mobile disco	3.33%	electronic
Gang starr	2.50%	Hip-Hop
Tosca	3.37%	electronic
Isis	2.77%	experimental
All time low	3.37%	rock
Above & beyond	2.13%	trance
Darkthrone	2.37%	black metal
Parkway drive	3.00%	metalcore
Bill evans	2.53%	jazz
Marisa monte	1.87%	brazilian
Pidzama porno	1.63%	polish
Covenant	2.47%	industrial
Jill scott	2.20%	soul
Håkan hellström	2.00%	swedish
The pigeon detectives	2.50%	indie
La oreja de van gogh	2.50%	spanish
Amethystium	1.87%	ambient
Sizzla	1.43%	reggae
Franz schubert	1.00%	Classical
ガゼット	1.13%	japanese
James newton howard	1.80%	Soundtrack
Teoman	0.67%	rock
Chris tomlin	0.83%	christian

Table 9: Representative artists, Boolean AP

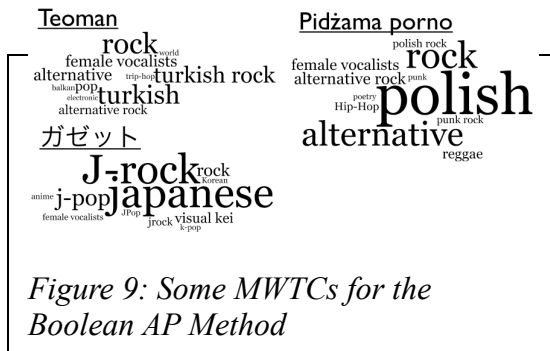


Figure 9: Some MWTCs for the Boolean AP Method

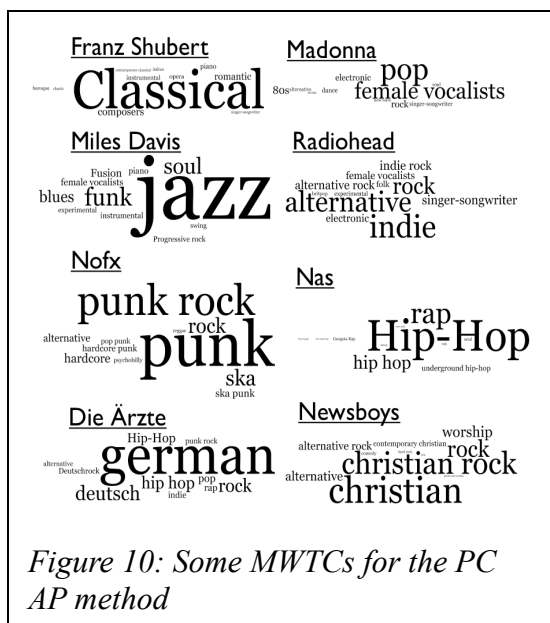


Figure 10: Some MWTCs for the PC AP method

3.4 Eclecticism level evaluation

In order to stimulate latent interests, the recommender system should propose artists belonging to Musical Worlds which are not already known to the user. Classical user-based or item-based approaches are not suitable for this purpose: the item-based approach is typically a conservative approach, if it is used to propose artists similar to the ones the user already likes, the artists will be probably part of a Musical World the user already knows; the user-based approach can add more diversification, considering artists that other similar users (neighbours) like and the served user does not know, but those artists can still be part of a Musical World the user already knows,

indeed, it is likely that, for the user's neighbourhood, the new artists that are more likely to be proposed (because they have high value playcounters for many neighbours) are artists similar to the set of artists the user already knows.

A critical step which needs to be completed in order to stimulate latent interests is therefore to understand which are the Musical Worlds a user already knows and likes. The easiest way to accomplish this task is to get, for each artist a user likes, the corresponding Musical World the artist belongs to and then consider a Musical World as “known” if the number of artists the user likes that belong to that Musical World is greater than a threshold min_ar_mw . Considering a Musical World as “known” even if the user likes just one of the belonging artists (and therefore set min_ar_mw to 1) would probably lead to distortions and false positive in the results; to be enough confident in classifying a Musical World as “known” we set min_ar_mw to 3 in our experiments. We called this approach *simple_mw_assignment*.

While the approach described above is reasonable and requires a small computational effort, we also tried a second approach which exploits the notion of exemplar introduced in paragraph 3.3.2. Since the exemplar should be an artist which well represents the artists belonging to the same Musical World, every artists whose similarity with an exemplar is over a threshold min_sim_ex can be considered part of the same Musical World, even if the clustering process put them in another one. We called this approach *overlapping_mw_assignment*. This approach, respect to the previous one, has two advantages:

1. While an artist, after the clustering process, is assigned to one and only one Musical World, it could be close enough to the exemplar of another Musical world to be reasonably considered part of this

second Musical World as well. In other words, it is possible to simulate an overlapping clustering.

2. While Affinity Propagation works well in this domain there could arise situations in which an artist is assigned to a Musical World even if the similarity with the corresponding exemplar is not high. This could happen because in the clustering configuration obtained there is not an exemplar that is very similar to the artist. Consider that this latter situation can arise even using an exact clustering approach, but it is more likely to happen considering the fact the Affinity Propagation is an approximate clustering method³⁸.

For an example of the first situation, the band *Pendulum* in the second clustering experiment has been classified in the musical world represented by the band *Radiohead*, the similarity between the two bands is 0.022, the band *Pendulum*, however, shows a similarity value of 0.022 even with the artist *nu:tone*, exemplar of another Musical World. The MWTCs of the two Musical Worlds are the followings:



Figure 11: Radiohead's Musical World

³⁸ An exact clustering method is a computational expensive clustering method that tries all the possible clustering configuration in order to choose the best



Figure 12: nu:tone Musical World

while the personal tag clouds of the artists *Pendulum* and *nu:tone*, retrieved from last.fm^{39,40}, are the following ones:



Figure 13: nu:tone last.fm tag cloud



Figure 14: Pendulum last.fm tag cloud

If we use the *simple_mw_assignment* approach, the fact the a user likes *Pendulum* will be considered just as an evidence that the user like the *Radiohead's* Musical World; if we use the *overlapping_mw_assignment*, instead, the fact will be also considered as an indicator that the user likes the

39 <http://www.last.fm/music/Nu%3ATone/+tags>

40 <http://www.last.fm/music/Pendulum/+tags>

3 - Proposed approach

nu:tone's Musical World; if we analyze the MWTCs and the artists' tag clouds, it is straightforward to realize that considering *Pendulum* also as part of the *nu:tone's* Musical World is correct since both the artists play *Drum and bass* music, and probably more reasonable than considering them as part of the *Radiohead's* Musical World.

For an example of the second situation, consider the band *Kumm*, assigned in our second clustering experiment to the Musical World represented by *Felix mendelssohn*. The similarity between the two artists is 0.004, very low; the low value is also explicable if we analyze the tag clouds of the two artists as retrieved from last.fm^{41,42}:

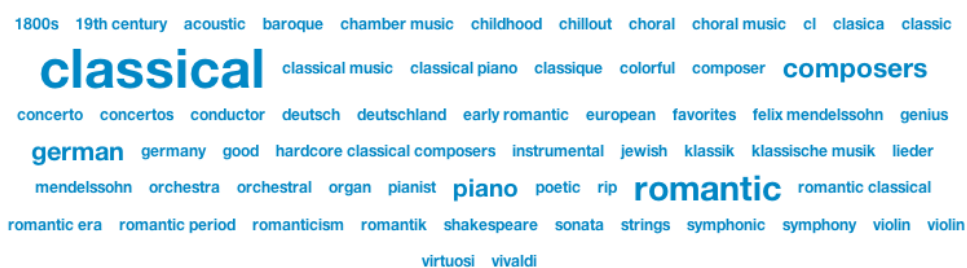


Figure 15: Felix Mendelssohn's last.fm tag cloud



Figure 16: Kumm's last.fm tag cloud

41 <http://www.last.fm/music/Kumm/+tags>

42 <http://www.last.fm/music/Felix+Mendelssohn/+tags>

The band *Kumm*, however, does not show any higher similarity with the other exemplars of our clustering, so although it is technically correct to assign the band to the *Felix mendelssohn's* Musical World, it would not be a good indicator of the fact that a user knows and likes that Musical World. For this reason, it is preferable to discard the “like” information about this band, because they are not a good indicator for assuming the like on a Musical World. This does not mean that this band is not similar to any of the other bands in our artists dataset: the band shows a high similarity value (> 0.1) for eight bands in the dataset. However, none of the similar bands has been chosen to be an exemplar; this could happen for two reasons:

1. the group of similar artists that include *Kumm* is too small and does not show enough peculiarity to be considered an independent Musical World, considering the number of the clusters obtained
2. it does exist an alternative and better (or at least as good as the current) clustering solution where *Kumm* belongs to a Musical World whose exemplar show an high similarity with the artist, but the Affinity Propagation clustering didn't find it.

We do not know which is the actual reason because we didn't apply an exact clustering algorithm to our dataset_30000 dataset.

A further alternative way to check if a user knows and likes a Musical World is to consider the similarity of the artists with the exemplars as a weight, and consider a Musical World as known and liked by a user if, with the artists he likes, he reaches a threshold score computed by adding the similarities of the artists liked with the exemplar of the Musical World. Even in this case, however, it would be necessary to consider only the artists having a similarities with the exemplar over a certain threshold, otherwise a user having lots of artists' likes can reach the score too easily even if the

artists are clearly not part of the Musical World. This approach has not been tested.

Calculating the number of Musical Worlds that a user knows and likes is useful not only to understand which Musical Worlds propose to him in order to stimulate his latent interests, but also to understand who, among the users, is more eclectic. Given the set of the Musical Worlds $MW(mw_1, mw_2, \dots, mw_n)$ and a function $userLikesMW(mw_i)$ that returns *true* if the user knows and likes the Musical World mw_i , the eclecticism level of a user is computed as:

$$\frac{|\{mw_i : userLikesMW(mw_i) = 1\}|}{|MW|} \quad (13)$$

Where a score 1 means that a user is very eclectic, knowing and liking all the Musical Worlds.

Even if we consider the index proposed as a valid indicator of eclecticism, it is important to remark that the eclecticism's level could be underestimated for three reasons:

1. The original dataset⁴³ has been built considering, for each user, the most played artists; for some users that approach could have led to discard some of them.
2. The original dataset considers the songs played and *scrobbled* to Last.FM; of course a user could scrobble just a portion of the music he listens.
3. The derived dataset contains information just about the top 30,000 artists and the dataset cleaning operations discarded some artists, as described in details in chapter 3.1

⁴³ <http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-360K.html>

It is also important to remark that, as explained in detail in chapter 1.2.1.6 , while it is easy to derive implicit positive feedback from playcounters information, detecting negative implicit feedback is hard; if a user does not like enough artists related to a Musical World, we assume that he does not know that Musical World and that suggest him the relative artists is valuable; the user, actually could know very well and dislike that Musical World. This problem is mitigated if we consider two additional details:

1. The way the (assumed) unknown Musical World is proposed (see chapter 3.6.3) could change the opinion the user already had about a Musical World;
2. As we will explain in chapter 4.2 , the interface of the radio application proposed provides a dislike button that the user can use to explicitly express his opinion about an artist (that can impact on the music played later).

3.5 “Mentor” approach

The process of suggesting Musical Worlds which are not already known to the user requires to solve two problems:

1. decide which one, among the unknown musical worlds, to suggest;
2. decide which artists belonging to the unknown musical world to suggest and create a playlist.

We will present the solution adopted for the second problem in chapter 3.6.3 ; for the first problem we propose, instead of choosing a random musical world, to exploit the knowledge of the most eclectic users. The idea behind this approach is borrowed by the real world, where music passionates help other people in their social network (e.g. friends) giving hints to discover new music. In order to help a user in broadening his horizons, in fact, other

users who already experienced this broadening process can give a better help than non-eclectic users: we therefore consider the eclectic users as mentors who can guide the other users in the discovering process. In order to exploit their knowledge we applied a user-based collaborative-filtering process, which differs from a traditional user-based collaborative-filtering process for two reasons:

1. Instead of taking into consideration the set of all the users to choose the neighbourhood, we consider just the most eclectic users (we call them “mentors”); this means that this approach could for example prefer, as neighbour for a user U_i , user U_j respect to user U_z even if $similarity(U_i, U_j) < similarity(U_i, U_z)$ if U_j is an eclectic user and U_z is not.
2. Instead of using artists as items in the collaborative filtering problem, we use MWs: since at this stage of the process we are trying to recommend a MW, it is more logical to compute similarities at a MW-level instead that at an artist-level, using aggregated data about a musical world instead of data about artists.

More formally, considering a set of users U and a set of musical worlds MUS , the entire process can be described as follows:

1. Extract, from the set of the users U , the set of the mentors UMN : they are the users who know the higher number of different MWs, and they can be chosen considering the users who know a number of MWs higher than α , where α is a parameter;
2. Create a matrix $M(|UMN| + 1, |MUS|)$ where each row i represents one of the $|UMN|$ mentors or the current user, each row j one of the $|MUS|$ Musical Worlds and each cell $M(i,j)$ how many artists of a musical world j the user i knows and likes;

3. Compute the similarities between the current user and each of the mentors, considering each time the user and a mentor as a vector of musical worlds and computing a Jaccard index as explained in equation 7;
4. Consider the subset UMN_s of the mentors set: the first k mentors who are most similar to the current user, where k is a parameter;
5. Create an ordered list MUS_P of musical worlds ordering the elements of MUS according to the preferences of the mentors belonging to UMN_s , where the most liked musical worlds will be at the top of the list and the less liked at the bottom.

The list MUS_P can then be used to choose which musical world propose first to the user, starting from the firsts of the list and avoiding musical worlds having zero preferences.

3.6 Playlist generation

3.6.1 Introduction

In this work we define a playlist similarly to how is defined in [Fields, 2011]; in particular, we consider a playlist as a sequence of songs that is produced to be listened in a given order, typically without interruption, by a user. While the number of songs contained in a playlist can vary, in this work we usually refer to a playlist as a list having a few to twenty songs, that represents the length typically associated with the term (think about mixtapes, albums, radio shows).

Since we worked on a artist similarity level and not on a song similarity level, in this work a playlist is actually treated as a sequence of artists; a constraint has been added at the implementation level in order to improve the playlist with diversity and make them more interesting: the same artist cannot be repeated in the same playlist.

After having decided which artists recommend to a user, the generation of the playlist, that means choose in which order play the artists, is itself a research problem. In this work, we faced two different playlist generation problems:

1. Generate a playlist having in input the artists the playlist should contain;
2. Generate a playlist having in input just the first and the last artist.

The latter is strictly related to our fifth subproblem (see chapter 2.3) and the solution proposed is presented in detail in chapter 3.6.3 , the former is related to the production of a playlist in “normal mode”(see chapter 4.4) and the solution proposed is presented in the following chapter (3.6.2).

3.6.2 Playlist generation, given in input the artists

A metric that can be used to evaluate the quality of playlist, given the songs, is the *cohesion* of a playlist [Fields, 2011], that is the average distance in a playlist between a song and the following one. This definition needs a concept of distance; having defined a method to compute artist's similarities (see chapter 3.2) and having assumed that similarities between artists can approximates similarities between songs, we can measure the distance *dist* between two songs *i* and *j* in a playlist as:

$$dist(s_i, s_j) = similMax - simil(art(s_i), art(s_j)) \quad (14)$$

where $art(s_i)$ is the artist of the song *i*, $simil(art_x, art_y)$ returns the similarities between two artists *x* and *y* using the boolean collaborative-filtering metric described in chapter 3.2.2) and *similMax* is the maximum value of similarity in the similarity scale proposed (in our case 1 for collaborative-filtering similarities and 100 for folksonomy-based similarities). In figure 17 a representation of the conversion between matrix and graph with an example is presented.

Simil (Charlie Parker, Miles Davis)

	Charlie Parker	Miles Davis	Metallica
Charlie Parker	1000	108	4
Miles Davis	108	1000	19
Metallica	4	19	1000

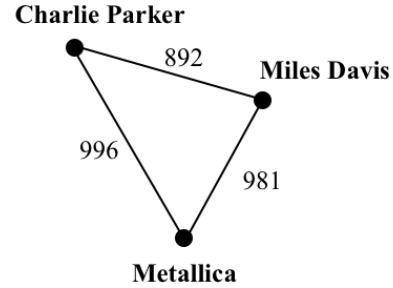


Figure 17: The Similarities Matrix S and the derived graph G (values are multiplied by 1000)

In order to solve the playlist generation problem, we adopted a graph search approach. We represented the similarity matrix S as a graph $G = (V, E, W)$ where:

- for each pair of $x, y \in V$, there is an edge $e(x, y)$
- the weight $w(x, y)$ of the edge is $dist(s_i, s_j)$

Having a set of n artists $Ap(a_1, a_2, \dots, a_n)$ we want to include in a playlist and having an artists graph G , finding the playlist with the best cohesion means finding a solution for the *Travelling salesman problem*, i.e. finding the shortest path Ps which covers all the artists. We first applied a simple brute force solution to the problem, trying all the possible permutation of the artists and and see which one is the shortest. To evaluate the technique, we used another graph $G_tag = (V, E, W)$ where:

- for each pair of $x, y \in V$, there is an edge $e(x, y)$
- the weight $w(x, y)$ of the edge is $dist_tag(s_i, s_j)$

where $dist_tag$ is the distance computed using the tag cloud similarity measure described in chapter 3.2.4.4 , that represent our ground truth, and

see how much, computing the distance using the graph G_{tag} , a random path Pr is longer (if it is) than the Ps path found using our G graph.

To test the approach, we performed the following preliminary experiment (see chapter 5, instead, for the main experimental results). We considered the set of the 3,000 most popular artists in our dataset (dataset_3000); from that dataset we randomly selected a set of artists Ap , we ordered the artists as explained above, obtaining a path Ps , and then we measured $diff$ as how longer, in percentage, (considering the graph G_{tag}) is the average distance between one artist and the following one in the the path of artists Pr respect to the ordered path Ps . We repeated the random extraction for 100 times and computed the mean of $diff$ and we conducted three different experiments considering 5 artists, 7 artists and finally 9 artists. In the following table a synthesis of the results.

	5 artists	7 artists	9 artists
mean(diff)	+4.83%	+6.91 %	+7.97 %

The results show that Pr paths provides, on average, a mean distance between an artist and the following one which is longer than the one provided by Ps paths; furthermore, it seems that the longer (in terms of number of artists) is the playlist, the higher is the difference. While the ordering algorithm allows an improvement in cohesion, the improvement is, in this first experiment, limited. This is due to the huge set from which we chose the artists: by choosing randomly from this set it is not unlikely to extract artists who do not share users and for this reason their distance is considered equals to $similMax$; in these cases, the ordering algorithm has not room for improve the random order.

In the second experiment we extracted all the artists from the same Musical World, in particular from the one represented by the artist “The Rolling Stones”. Here are the results:

3 - Proposed approach

	5 artists	7 artists	9 artists
mean(diff)	+13.26%	+19.19 %	+20.93 %

The improvement in the second experiment is more evident, because being in the same Musical World, it is less likely to face cases in which artists do not share users. Even in this second experiment the improvement still seems positively correlated with the number of artists selected.

In the third experiment we extracted the artists from two musical worlds, and in particular we chose the “Simian mobile disco” MW and the “Tosca” MWs, which, from the stylistic point of view, could be considered close (see the MWTCs in figure 18 and 19). The improvement is even more evident in this third example, and still seems positively correlated with the number of artists selected.

	5 artists	7 artists	9 artists
mean(diff)	+22.40%	+27.22 %	+32.62 %



Figure 18: The musical world represented by Tosca



Figure 19: The musical world represented by Simian Mobile Disco

3.6.3 Playlist generation, given the first and the last artist

3.6.3.1 Introduction

Choosing which artists, among the ones belonging to the proposed (and unknown) musical world, suggest to the users is not a trivial task, especially if the similarity of the artists with the artists explicitly liked by the user is very low. The risk is to propose music which is too far from the current user's taste. In order to be trusted by the user, a recommender system requires to firstly propose some items which will be most probably closer to the user's taste and then take the risk to propose some less obvious items that the user, at a first extent, can hardly recognize as part of his current interests.

The idea proposed in this work, called “Musical World Trip” is to create a playlist which starts from one or two songs by artist liked by the user, similar to them or borderline respect to the user's taste and ends with a song by the artist considered as exemplar of the musical world proposed. The intermediate songs should be by artists who are firstly similar to the first (or second, if two) artist and then tend to be gradually more similar to the end artist and less to the start artist; in this way the user can be exposed gradually, as time passes, to the new musical world. Note that, following this approach, some of the artists included in a playlist belong to the target and unknown musical world (at least one, the last) and some others could belong to other musical worlds.

3.6.3.2 The Artists' graph

In order to get the “Musical World Trip” playlist, we, again, represented the similarity matrix S as a graph $G_2 = (V, E, W)$ where:

- for $x, y \in V$, there is an edge $e(x, y)$ if $s(x, y) > k$
- the weight $w(x, y)$ of the edge is $dist(s_i, s_j)$

The only difference respect to the G graph presented in the previous chapter is that in G_2 two artists are connected only if their distance is below a threshold k . We set k to 0.11.

Once you got the graph G , the task of getting a gradual path from the artists A_1 to the artist A_2 can be approximated trying to get the shortest path which connects A_1 and A_2 : since only similar artists are connected, we are sure that the transition from the first to the last song is smooth.

3.6.3.3 Application of the Floyd's Algorithm

The computation of the shortest path can be expensive to execute on-line, so we decided, for the recommendation radio experiment (see chapter 4), to compute all the paths between all the 30,000 artists taken into consideration (dataset_30000) off-line and to store the paths in a relational database. In particular, an implementation of the Floyd's algorithm [Floyd, 1962] has been used.

The Floyd algorithm is an algorithm for finding shortest paths in a weighted graph; it returns the distances for all the shortest paths among all the vertices in a graphs and has a complexity of $O(n^3)$, where n is the number of vertices in the graph. The original algorithm just returns the distances and not the paths but applying a simple and known modification it is possible to also get the information about the paths themselves without affecting the computational complexity. The pseudocode of the final algorithm used is presented in the followings.

Given the matrix $D[n][n]$ containing the weights of the edges among all the n vertices (or k , where k is a number \gg of the maximum weight in the dataset, if a pair of vertices is not connected) and a matrix $X[n][n]$, initialized as -1 for all the elements, the following pseudocode:

3 - Proposed approach

```
for (z = 0; z < n; z++) {
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            if (D[i][z] + D[z][j] < D[i][j]) {
                D[i][j] = D[i][z] + D[z][j];
                X[i][j] = z;
            }
        }
    }
}
```

returns a modified matrix $D[n][n]$ where each element $D[n_i][n_j]$ contains the distances d (sum of the weights of all the edges) between vertex i and vertex j (-1 if there is no paths available from i to j) and a modified matrix $X[n][n]$ where each element $X[n_i][n_j]$ contains the intermediate vertex to visit for going from i to j (-1 if it is the last intermediate vertex before j). The matrix X can then be easily recursively traversed using a known approach (like the one we present in pseudocode below) in order to get the ordered list of vertices that are in the path from a vertex i to a vertex j . Here is in pseudocode a function *print_path* which, having in input a vertex i and a vertex j can print the path from i to j .

```
print_path (i, j)
{
    next = X[i][j];
    if (next == -1){
        return;
    }
    print_path(i,next);
    print(next);
    print_path(next, j);
}
```

3.6.3.4 Examples

In this paragraph we present some examples of the Musical World Trip technique. In particular, some paths between well-known artists, computed using the technique explained in the previous chapter, will be illustrated.

For each example, we show the intermediate artists in the path, a synthetic version of the tag clouds, containing the most important tags and the similarity with the destination artist (the last one in the path). Consider that in the real implementation for Mentor.FM (see chapter 4) the system tries to build a playlist according to the algorithm just explained but for some reasons (for example there could be a technical problem in playing the songs of an artist) an artist could be not available so in the final playlist there could be some “holes” in the sequence of artists.

In the first example, we computed the path between two artists whose similarity is quite high: the rock band “Coldplay” and the classic rock band “The Rolling Stones”, whose similarity value is 0.075. The path includes just one band, “The Beatles”.

In the second example, we still started from the the rock band “Coldplay” but this time we tried to reach a further artist in the graph, the electronic band “The Prodigy”, whose similarities with “Coldplay” is 0.055; in this example we need three artists, “Radiohead”, “Massive attack” and “The chemical brothers”. Looking at the tag cloud, it is clear that, during the path, the music genre slightly move from rock to electronic: the word “electronic” is in fact absent in the first tag cloud and then becomes bigger and bigger while the destination is closer.

In the third example, we computed a longer path, from the electronic band “Simian mobile disco” to “Aerosmith”, the similarity between the two bands is very small, 0.002: not only the band genre is very different but also they achieved success in a different period, therefore they probably share a very small portion of their audience. In this third case, seven artists are needed to reach the final one. Again, from the tag cloud we can see that the music genre slightly moves from electronic to rock, and we can argue that the band “Massive attack” works as a glue to keep together artists playing different music genres.

In the fourth example, we computed a path from the punk rock band “Social distortion” to the rap band “Wu-tang clan”; the similarity between the two bands is small, 0.005 and the music genre is very different. In this fourth case, seven artists are needed to reach the final one. Again, from the tag cloud we can see that the music genre moves from punk/punk rock to hip-hop/rap; only one band, Linking park, has not one of those tags in its tag cloud, we can argue that, since it is well-known to be a band playing a crossover genre between rap and rock, the band works as a glue to keep together artists playing different music genres.

FIRST EXAMPLE

1) Coldplay (last sim. 0.075)



2) The Beatles (last sim. 0.185)



3) The Rolling stones



SECOND EXAMPLE

1) Coldplay (last sim. 0.055)



2) Radiohead (last sim. 0.056)



3) Massive Attack (last sim. 0.100)



4) The chemical brothers (last sim. 0.156)



5) The Prodigy



THIRD EXAMPLE

1) Simian mobile disco (last sim. 0.002)



2) Justice (last sim. 0.006)



3) Daft Punk (last sim. 0.026)



4) The chemical brothers (last sim. 0.017)



5) Massive attack (last sim. 0.014)



6) Radiohead (last sim. 0.028)



7) The Beatles (last sim. 0.061)



8) Queen (last sim. 0.121)



9) Aerosmith



FOURTH EXAMPLE

1) Social distortion



2) Rancid (last sim. 0.010)



3) nofx (last sim. 0.010)



4) Rise against (last sim. 0.005)



5) Sum 41 (last sim. 0.003)



6) Linkin park (last sim. 0.005)



7) Eminem (last sim. 0.037)



8) Nas (last sim. 0.114)



9) Wu-tang clan



In some cases the algorithm creates a path that is optimal (from the length point of view) but that at some point goes through a path that moves too far from the destination instead of getting closer. An approximated representation on a two-dimensional space of this behaviour is given in figure 20 where, in order to go from artist A to artist B, which are not connected because their similarity is under the threshold 0.11, a path through the artists C, D, E, F, G, H, I is walked.

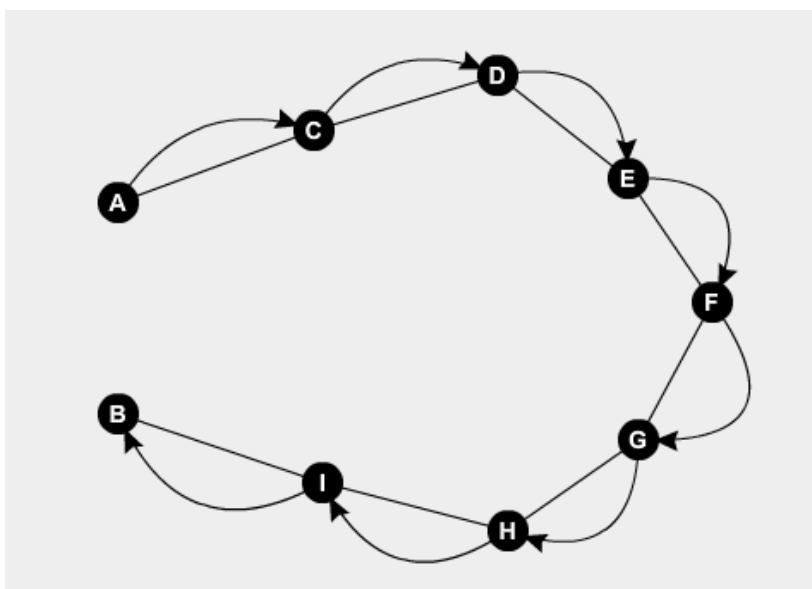


Figure 20: Path between artists A and B

A real example is the path from “U2” to “Norah jones”; the similarity between the two artists is 0.071, that means their audience is not very different, but still the value is under 0.110, so they are not connected in the graph. Here is the ordered list of artists that the algorithm includes in a playlist that starts with a song by “U2” and ends with a song by “Nora jones”. The playlist is very long (36 songs) so, instead of including for each artist his tag cloud, we wrote between parenthesis the first two (and most used) tags of the tag cloud; the similarity of the artist with the destination

artist is also included after the tags. Note that a similarity value ≤ 0.003 is approximated to 0.000. It is easy to see that the playlist go through several music genres and styles and that, starting from the second artist, the similarity with “Nora jones” decreases until the 32th artist, when starts increasing. While this playlist has a good cohesion, guaranteed by the constrain we imposed to the graph edges, it is not the best approach, from the musical point of view, to walk through so many MWs and firstly getting far from the destination, before reaching the last artist in the playlist.

- 1) U2 (rock, classic rock, sim. 0.071)
- 2) Red hot chili peppers (rock, alternative rock, sim. 0.049)
- 3) Metallica (thrash metal, metal, sim. 0.024)
- 4) Slayer (thrash metal, metal, sim. 0.005)
- 5) Napalm death (grindcore, death metal, sim. 0.000)
- 6) Nasum (grindcore, death metal, sim. 0.000)
- 7) Regurgitate (grindcore, goregrind, sim. 0.000)
- 8) Dead infection (goregrind, grindcore, sim. 0.000)
- 9) Agathocles (grindcore, mincecore, sim. 0.000)
- 10) Extreme noise terror (grindcore, Crust, sim. 0.000)
- 11) Aus-rotten (Crust, anarcho-punk, sim. 0.000)
- 12) Nausea (Crust, crust punk, sim. 0.000)
- 13) Amebix (Crust, crust punk, sim. 0.000)
- 14) Anti cimex (d-beat, Crust, sim. 0.000)
- 15) Kaaos (hardcore punk, punk, sim. 0.000)
- 16) Maho neitsyt (punk, hardcore punk, sim. 0.000)
- 17) Häiriköt (punk, ramopunk, sim. 0.000)

- 18) Barnyard ballers (psychobill , rockabilly, sim. 0.000)
- 19) Os catalépticos (psychobilly, power psychobilly, sim. 0.000)
- 20) Demented are go! (psychobilly, rockabilly, sim. 0.000)
- 21) Stray cats (rockabilly, psychobilly, sim. 0.000)
- 22) Eddie cochrane (rockabilly, Rock and Roll, sim. 0.000)
- 23) Ricky nelson (oldies, rockabilly, sim. 0.000)
- 24) Bobby vee (oldies, 60s, sim. 0.000)
- 25) Neil sedaka (oldies, 60s, sim. 0.000)
- 26) Pat boone (oldies, 60s, sim. 0.000)
- 27) Guy mitchell (50s, country, sim. 0.000)
- 28) Billy vaughn (instrumental, orchestra, sim. 0.000)
- 29) Ray anthony (jazz, swing, sim. 0.000)
- 30) Kay starr (jazz, female vocalists, sim. 0.000)
- 31) Lena horne (jazz, female vocalists, sim. 0.000)
- 32) Dinah washington (jazz, female vocalists, sim. 0.021)
- 33) Sarah vaughan (jazz, female vocalists, sim. 0.027)
- 34) Ella fitzgerald (jazz, female vocalists, sim. 0.071)
- 35) Diana krall (jazz, female vocalists, sim. 0.115)
- 36) Norah jones (jazz, female vocalists)

Since we deleted the edges whose weight was over a certain threshold, the maximum distance between an artists and the following one in the playlist is fixed and for this reason where the graph is not enough dense the algorithm could produce a long playlist like the one shown above. An alternative approach to the problem could be an algorithm which does not require a

maximum fixed distance between two artists but, instead, a maximum amount of “credit” to spend relaxing the constraint if necessary. Back to figure 20 the algorithm would add artist B immediately after artist A if the “credit” is enough, avoiding to add to the path the artists C, D, E, F, G, H and I. We didn't implement this alternative approach; however, we should highlight that the playlists produced during the last experiment with Mentor.FM (see chapter 4 and paragraph 5.2 for details) suggests that in a real-world scenario, paths like the one described above should be produced quite rarely, at least in combination with the technique we used to choose the final artist. Just 8 playlists (out of 89 produced) had a total number of artists greater than 10 (3 having 11 artists, 1 having 12, 3 having 13 and 1 having 22).

4

Application: mentor.fm

4.1 Introduction

To test the approach proposed, a real personalized recommendation radio application, Mentor.fm, has been developed. The application was Web-based and available at the URL <http://mentor.fm>. A user, after having logged into the application using his Facebook account, can immediately start listening to some music which Mentor.fm recommends for him.

7digital⁴⁴, a media delivery company based in London and operating globally, provided free access to their songs file archive and to their streaming servers for a limited number of users and a limited period; this allowed to experiment the radio in a real-world scenario.

4.2 Interface and features

The user was provided with a classical player interface having play, pause, stop and skip (to the next song) buttons. A playing bar allows the user to see how much of the song has been played and how much has been buffered and also to move back and forward to a precise part of the song. The volume control and a mute button completed the player interface.

In addition to a classic player interface, two buttons, a “thumbs up” button and a “thumbs down” button allowed the user to express his vote

⁴⁴ <http://7digital.com>

about the current song. The system kept track of the user's activity so that for each song, we knew if the user either:

- just listened to the song or
- just skipped the song or
- liked (“thumbs up”) the song or
- liked (“thumbs up”) + skip the song or
- disliked (“thumbs down”) or
- disliked (“thumbs down”) + skip the song.

The user's activity recorded was used for the evaluation of the experiment, as explained in chapter 5 .

Finally, a switch “surprise me!” allowed the user to select between the “normal” mode and the “surprise me!” mode; all the details about the two playing modalities are described later in this chapter.

While a typical Web radio usually shows the information about the current song played (song's title and artist's name), we chose to hide this information to the user. The first aim of the experiment was to understand the most effective way to propose music apparently far from users' taste and we thought that very often the opinion on a song can be distorted by the opinion (even from the social point of view) a user have on the artist. For this reason, we thought that hiding the information let the user focus just on enjoying the music and could make the rating activity more neutral. We received, however, many negative feedback messages by the users about this choice so after some weeks we finally decided to show the information. For most of the duration of the experiment described in paragraph 5.2.2 , therefore, that information was available to the user.

In figure 22, 23 and 24 it is possible to see the three main screens of the application: the starting screen, the playlist-generation screen and the play screen.

4.3 Technologies

The application has been developed using the PHP⁴⁵ programming language as a server-side language, while the client interface has been developed using HTML, CSS and Javascript. The data was stored in a relational MySQL⁴⁶ database.

In particular, the jQuery⁴⁷ javascript library has been used to simplify the client-side programming, particularly the handling of DOM⁴⁸ elements and the implementation of Ajax⁴⁹ techniques. Most of the Ajax techniques have been used to store information during the listening time: every time a new song starts, the user skips to the next song or click on the like or on the dislike button, the action is recorded into the database; in order to avoid the re-loading of the Web page (and consequently the interruption of the audio flux) an asynchronous (Ajax) call to a PHP script which handles the storing operation was the best and way to preserve a good user experience.

For the player interface, we used jPlayer⁵⁰, a free and open source media library written in Javascript and available as jQuery plugin.

The application made large use of the Facebook development platform⁵¹, first of all during the authentication procedure. We decided to allow login just from users having a Facebook account, through the Facebook Authentication⁵² procedure, for several reasons:

45 <http://www.php.net>

46 <http://www.mysql.com>

47 <http://jquery.com/>

48 http://en.wikipedia.org/wiki/Document_Object_Model

49 [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))

50 <http://jplayer.org>

51 <https://developers.facebook.com/>

52 <https://developers.facebook.com/docs/authentication/>

1. Easiness: it is quite common, especially for young people, to keep the Facebook page opened in a browser window while doing other Web activities; if a user is already logged in Facebook, it needed just one click to login in Mentor.fm, no username and password need to be provided.
2. Access to the user's music preferences: Facebook users can express their preferences about musicians (in figure 21 you can see a screenshot of the music section in a Facebook profile) and through the Facebook APIs it is possible to access the list of musicians the users “liked” on the social network; this is a precious information, providing a starting set of explicit preferences in input.
3. Popularity: having more than 800 million registered users, it is hard to find people, especially if interested in Web music platforms, who do not have a Facebook account.

The Facebook APIs provide the names of the artists liked by a user, together with a unique ID number assigned to the artists by Facebook. Our original Last.fm dataset provides the names of the artists played as well, together with a (different) unique ID number. In order to link the two dataset we used the name of the artist; while the solution is of course not perfect (there are different artists having the same name) we think it was a good approximation. The two datasets needed to be linked also to the 7digital archive to retrieve the songs to play; even in this case, the IDs vocabularies were different so we used the name of the artist as a link. In details, we used the artists/search method⁵³ of the 7digital APIs to check if the artist we wanted to play was available in the 7digital archive and, if available, we used the ID returned as input for the artists/toptracks⁵⁴ method to get the most popular songs by an artist. The 7digital ID of the artists and all the

⁵³ <http://api.7digital.com/1.2/static/documentation/7digitalpublicapi.html#artist/search>

⁵⁴ <http://api.7digital.com/1.2/static/documentation/7digitalpublicapi.html#artist/toptracks>

information about the songs are then stored to avoid unnecessary calls to the APIs the next time the artist needs to be played.

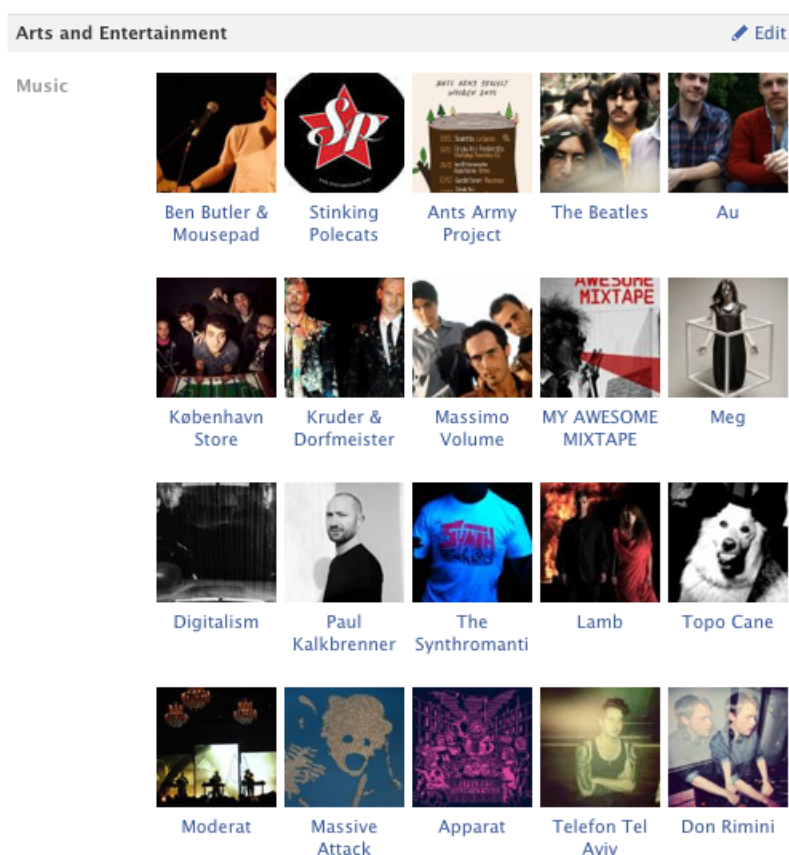


Figure 21: Facebook profile, screenshot of the music section



Figure 22: Mentor.fm starting page

4 - Application: mentor.fm

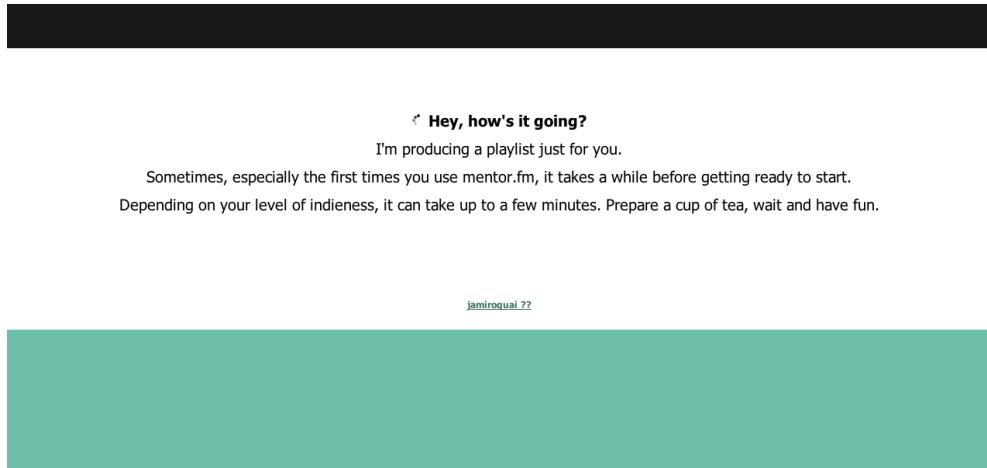


Figure 23: Mentor.fm playlist creation page



Figure 24: Mentor.fm playing screen

4.4 Playlists creation and modalities

The radio had two play modalities: “normal” and “surprise me!”, both the modalities can work in two different ways according to the profile assigned to the user (see chapter 4.5 for further details) so there are actually four different playlist creation modalities: normal playlist (random profile), normal playlist (dj profile), surprise me! playlist (random profile) and surprise me! playlist (mentor profile).

4.4.1 Normal playlist, random profile

In this modality Mentor.fm tries to create a twenty songs playlist where the first five songs are by artists explicitly liked by the users via Facebook and the other fifteen songs are by artists similar to the ones explicitly liked, where the similarities are computed using the algorithm described in chapter 3.2.2 .

It is important to highlight that, while both the “thumbs up” and “thumbs down” buttons do not affect the current playlist, the buttons affects the future playlists: if a user dislikes a song, that song will not be included anymore in future playlists; furthermore, if a user dislikes two different songs by the same author, the vote is considered as related to the artist and all the songs belonging to that artists will not be included anymore in future playlists. A similar principle is applied to the like activity: if the user likes two different songs of the same artists, the artist becomes part of the explicitly liked artist, as it was an artist liked on Facebook. While deciding how to exploit the “dislike” inputs was clear since the beginning, the choice about the “like” inputs was not trivial: since we expected to use the likes as an evaluation of the algorithm, using them to also affect the input (the seed artists) of the algorithm could lead to interpretation problems, for this reason at the beginning of the experiment the “likes” didn't affect the playlist generation. We then realized that this could affect the user experience: a user

expect a positive feedback should have an effect on the radio, we therefore decided to change the approach.

4.4.2 Normal playlist, dj profile

In this modality Mentor.fm tries to create a twenty songs playlist exactly as in the previous modalities but finally reordering the songs found according to the similarities of their artists, in order to have the maximum possible playlist cohesion. The approach used was the one explained in chapter 3.6.2 and based on the travelling salesman problem; however, since computing the optimum path would have been not feasible considering twenty artists, a suboptimal solution has been introduced: only the five “seed” artists have been ordered, while all the other artists have been placed right after the seed artist they are similar to, considering that, if our algorithm propose an artist A_i as similar to an artist A_j the distance between the two should be low.

4.4.3 Surprise me! playlist, random profile

We tried different approaches for this modality but in the one we used for the final experiment described in paragraph 5.2.2 Mentor.fm created a fifteen songs playlist played by fifteen different artists extracted randomly from our dataset_30000 dataset.

4.4.4 Surprise me! playlist, mentor profile

In this modality Mentor.fm tries to create a playlist which starts with one or two artists who tend to be borderline respect to the user's taste i.e. similar to one of the artists the user already like but not included in the top 10 similar artists; the algorithm favors unexpected artists (see 5.2.1 for details about unexpectedness) but the unexpectedness is not guaranteed. The playlist continues with songs by artists who tend to be, gradually, closer to the most representative artist of the target cluster. The technique, including the choice of the MW to propose and the choice of the intermediate artists, is based on

the approach presented in chapters 3.4 3.5 and 3.6.3 ; however, the mentors are extracted from our dataset_30000 dataset, not from the Mentor.fm users, because the first is much richer. The playlist does not have a fixed number of artists as in the first three modalities.

4.5 Assignment of user profiles

Each user, at the first connection with Mentor.fm and, has been randomly assigned, for each of the two play modalities, with one of the two available profiles. Four different users profile combination were therefore available:

1. Normal playlist: random, Surprise me! Playlist: random;
2. Normal playlist: random, Surprise me! Playlist: mentor;
3. Normal playlist: DJ, Surprise me! Playlist: random;
4. Normal playlist: DJ, Surprise me! Playlist: mentor.

Sometimes, for the Surprise me! mode, we stopped (during the subscription) the random assignment and assigned just a specific profile in order to balance the activity of the two profiles; for the Normal mode, at some point we stopped (during the subscription) the random assignment and assigned just the DJ profile because we were not anymore interested in the Random vs. DJ comparison.

Users didn't know about this random assignment, they just knew there were two playlist mode: normal, which is supposed to play songs close to the user's taste, and “Surprise me!”, which is supposed to propose music apparently far from the user's taste and that Mentor.fm tries different algorithms in order to understand which one is the best.

A small group of users has been registered as “developer users”; they knew that the “Surprise me!” mode has two modalities and for a while they had an additional control to switch from one to the other. They were not considered part of the experiment, they have been asked to help the ongoing

development of Mentor.fm giving feedback about their experience with the application.

5

Evaluation results

Note: part of the paragraph 5.1 is extracted from our previous work:

- Tacchini, E. and Damiani, E. 2011. What is a musical world? An affinity propagation approach. In Proc. of the 1st ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies.

5.1 Evaluation of Artists' Classification

5.1.1 Metrics

The evaluation of a clustering algorithm can be performed in many different ways, according to different notions of cluster quality; in this work, as discussed, it is interesting to analyze the semantics of clusters.

Discovering hidden semantic selection criteria underlying a cluster of artists, however, is not a trivial task; while sometimes, especially when dealing with clusters containing just artists of a very specific genre, it is easy to discover the semantic selection criteria that justify the creation of a cluster of artists, sometimes discovering such a hidden criteria is not a trivial task: artists in the same cluster could have the same geographic origin, the same production process (e.g., independent distribution), a similar production target (e.g. music for soundtracks) or other factors related to the cultural and social environment where the relative songs were composed.

Folksonomy can be considered a very useful classification system, because through a list of tags it is easy to express different semantic facets. We therefore validated our clusters performing the analysis of the tag clouds corresponding to the artists in each cluster, as explained in details in the followings.

Using the last.fm API `artist.getTopTags`⁵⁵ we retrieved the tag cloud for each of the 3,000 artists in the *dataset_3000* dataset and computed the Musical World Tag Cloud (an aggregated tag cloud, see chapter 3.3.2.3) for each musical world produced by the different clustering techniques, in order to intuitively and graphically understand the composition of the MW.

To analytically understand if a MW was coherent, we calculated how much the tag clouds of the related artists were similar, using the *tagsim*, *wordsim* and *textsim* indexes presented in chapter 3.2.4.4 and following these steps:

1. Compute, for each artists couple in a musical world, the similarity using *tagsim*, *wordsim* and *textsim*;
2. Calculate the mean similarity of a MW by adding together the similarities computed in the previous step and dividing by the number of artists' pairs;
3. Compute a final score for each clustering technique by calculating the mean and the weighted mean of the 36 values got in the previous step, using as a weight the size of each cluster (in terms of number of artists).

We finally obtained six scores: values, for the indexes *tagsim*, *wordsim*, *textsim*, *tagsim_w*, *wordsim_w* and *textsim_w*, where the last three are the indexes obtained using the weighted mean.

⁵⁵ <http://www.last.fm/api/show?service=288>

5.1.2 Results

In table 10 we present the performance of three methods (K-means, Affinity Propagation using boolean similarity and Affinity Propagation using Pearson Correlation similarity) according to the six indexes proposed (*tagsim*, *tagsim_w*, *wordsim*, *wordsim_w*, *textsim* and *textsim_w*). Since *textsim* is computationally very expensive considering the long text we have to deal with (the complexity is $O(N^3)$ where N is the length of the longest string) we computed this index just for four cases; it is important to remark, however, that this index is not as representative of tag clouds homogeneity as the others ones. The performance has also been compared with four random clustering approaches:

1. *rnd_eq*: randomly assign each artists to a cluster, keeping similar the size of the clusters;
2. *rnd_k*: randomly assign each artist to a cluster, keeping the same cluster size obtained by the K-means method;
3. *rnd_b*: randomly assign each artist to a cluster, keeping the same cluster size obtained by the Boolean Affinity Propagation method;
4. *rnd_pc*: randomly assign each artist to a cluster, keeping the same cluster size obtained by the Pearson Correlation Affinity Propagation method.

Analyzing the results, it is clear that Affinity Propagation consistently outperforms K-means. In particular, PC AP performs better than Boolean AP for the *tagsim*, *wordsim* and *textsim* indexes, however Boolean AP is slightly better than PC AP for the weighted version of the indexes: *tagsim_w*, *wordsim_w* and *textsim_w*. PC AP seems to perform better on average, however, it seems hard to say which one of the two AP methods should be adopted, the suitability and therefore the choice is probably dependant on the context.

	tagsim	tagsim_w	textsim	textsim_w	wordsim	wordsim_w
rnd_eq	4.04	4.04	16.94	16.95	5.23	5.23
rnd_k	4.05	4.05			5.16	5.20
k-means	14.08	15.07	27.48	28.63	15.68	17.00
rnd_b	4.00	3.99			5.22	5.21
Bool AP	20.43	20.00	34.24	33.67	21.79	21.65
rnd_pc	4.11	4.05			5.11	5.21
PC AP	22.04	19.66	36.13	33.41	23.66	21.21

Table 10: Evaluation of artists' clusters

5.2 Evaluation of the surprising suggestions

5.2.1 Metrics

We present here a serendipity measure, the *serendipity cost* index, that can be easily used in a real-world music recommender radio system and maybe in some other domains. The idea is to evaluate, on average, how expensive is, for a user, to get a serendipitous recommendation in term of bad recommendations.

Firstly, a formal notion of *unexpected* suggestion is proposed. We started from the definition of [Murakami et al., 2008], where unexpectedness is measured as the deviation from the result provided by a primitive prediction method, and we tried to go further: since people are nowadays used to recommender systems, we define unexpectedness as the deviation from the results provided by a regular recommender systems. In our case, we use the normal mode of Mentor.FM (which can be considered a TOP-10 items based recommender systems) as our benchmark to decide if a recommendation is expected or not. In other words, a recommendation is

considered unexpected if the relative artist is not one of the artists the user “liked” on Facebook or one of the TOP10 similar artists (i.e. the artists the normal mode could recommend).

It is important to highlight that the time is also taken into consideration: since the taste and the eclecticism of a user evolve in time, a suggestions provided at time t_0 is considered unexpected only if it had the properties described above at time t_0 . Furthermore, the feedback the users provided on Mentor.FM using the like buttons are also taken into consideration: two songs belonging to the same artists “liked” on Mentor.FM are considered as a Facebook like on the artist.

Having explained formally which recommendations can be considered unexpected, in the following we formally explain which recommendations can be considered serendipitous and how to calculate the *serendipity cost* index. Let S be the set of all the triples (song, user, time) played by a music recommendation radio in a defined period of time, $SL \subseteq S$ the subset of the S recommendations which the users liked and $SD \subseteq S$ the subset of the S recommendations which the users disliked; let $S_2 \subseteq S$ be the subset of S containing triples whose songs played could be considered unexpected suggestion for the user; let $SL_2 \subseteq S_2$ the subset of the S_2 recommendations which the users liked. The *serendipity cost* of the recommender system can be expressed as the number of bad (disliked) recommendation the users have to afford in order to get a serendipitous recommendation:

$$serendipity\ cost = \frac{SD}{SL_2} \quad (15)$$

This measure takes into consideration both the components traditionally associated with serendipity: unexpectedness and satisfaction (see chapter 2.2) and also takes into consideration the related cost.

Since a typical music recommender system radio provides a user interface having like/dislike controls, we can exploit that information to compute the composition of the subsets SD and SL_2 . We also wanted to measure, as a precision indicator, the total cost of the algorithms, taking into consideration all the *likes* received as a feedback and not only the ones related to unexpected songs; we used the same idea:

$$total\ cost = \frac{SD}{SL} \quad (16)$$

The *total cost* can be seen as the number of bad (disliked) recommendation the users have to afford in order to get a very good (liked) recommendation. While the total cost, for our purposes, is not as crucial as the serendipity cost, we think that it is important to keep it low to avoid the user stops using the recommender system.

5.2.2 Results

We present here the results of the experiment conducted using our music recommendation radio Mentor.fm (see chapter and in particular 4.4): the main goal was to compare the performance of the *surprise me!* mentor algorithm to the performance of the *surprise me!* random algorithm. We think that the only two approaches found in literature that could be compared with our *surprise me!* mentor, because of the same aims, are [Oku, K. and Hattori, F. 2011] and [Abassi et al., 2009]; however, on our approach the time is an important factor and in a recommendation list (i.e. playlist) the order of the items does matter, for this reason, an accurate evaluation of the *surprise me!* mentor on a static rating dataset like the MovieLens Data Set used by both the works would not be possible. On the other hand, evaluating [Abassi et al., 2009] using our Mentor.FM radio would have meant an important effort of adaptation and implementation;

considering also the risk of concluding the experimentation without having enough data for the comparison (the users would have been divided in three groups instead of two) we therefore decided to consider that evaluation as a future work, together with a comparison with a standard user-based recommender system, which should be more “serendipitous” than an item-based recommender system. Regarding [Oku, K. and Hattori, F. 2011], the work is so recent (presented in late October 2011) that it would not have been even possible an evaluation of the technique in our radio.

The radio run for several weeks, however, for most of the time the final version of the mentor algorithm was not ready yet; the results presented are therefore referred to an experiment which lasts about 40 hours and started when the final idea could be considered ready.

Some data about the experiments is presented in table 11, together with the results about the likes/dislikes recorded. To make the data more accurate, we applied a filter: from the total number of songs played in normal mode, we discarded the ones belonging to playlists partially or entirely generated by the Last.FM recommendation APIs: this make the algorithm better comparable with the surprise modes, which does not use any external recommender as the normal mode does for artists who does not belong to the dataset_30000 set⁵⁶.

⁵⁶ For these artist, the normal mode use the APIs of last.fm to get similar artists

5 - Evaluation results

Users who used the radio during the experiment	168
Users having a surprise <i>random</i> profile (out of 168)	90
Users having a surprise <i>mentor</i> profile (out of 168)	78
Users having a normal <i>random</i> profile (out of 168)	7
Users having a DJ <i>random</i> profile (out of 168)	161
Users who used the radio in normal mode during the experiment	162
Users who used the radio in Surprise me! Mode during the experiment (random profile, mentor profile)	81 (48, 33)
Total songs played	1,654
Total songs played in <i>normal</i> mode	1,188
Total songs played in <i>normal</i> mode (filtered)	978
Total songs played in <i>surprise me!</i> mode	466
Total songs played in <i>surprise me!</i> mode random	228
Total songs played in <i>surprise me!</i> mode mentor	238
Number of LIKE recorded in <i>normal</i> mode	204 (20.86%)
Number of DISLIKE recorded in <i>normal</i> mode	93 (9.51%)
Number of LIKE recorded in <i>surprise me!</i> mode random:	22 (9.65%)
Number of DISLIKE recorded in <i>surprise me!</i> mode random	61 (26.75%)
Number of LIKE recorded in <i>surprise me!</i> mode mentor	30 (12.61%)
Number of DISLIKE recorded in <i>surprise me!</i> mode mentor	31 (13.03%)

Table 11: General data and like/dislike results

In table 12 we present the details about the number of songs we considered unexpected/expected and the number of “likes” recorded on unexpected songs. If, at the time the song was played, the relative user had not expressed any “like” preference, or the procedure of importing the “likes” from Facebook was not over yet, it was not possible to say if they were unexpected or not; the data about these songs were not considered when we computed the serendipity cost measure.

Normal mode	
Number of expected songs	685
Number of songs for which the expectedness is unknown	290
Number of unexpected songs	3
Number of LIKES received on unexpected songs	2
<i>Surprise me!</i> mode random	
Number of expected songs	0
Number of songs for which the expectedness is unknown	87
Number of unexpected songs played	141
Number of LIKES received on unexpected songs	15
<i>Surprise me!</i> mode mentor	
Number of expected songs	56
Number of songs for which the expectedness is unknown	11
Number of unexpected songs	171
Number of LIKES received on unexpected songs	15

Tabella 12: Results about unexpectedness and likes recorded on unexpected songs

In table 13 the total cost and serendipity cost indexes are presented for each of the three algorithms.

	Normal	Surprise me! Random	Surprise me! Mentor
Total Cost	0.46	2.77	1.03
Serendipity Cost	20.50	2.60	1.60

Table 13: Total cost and Serendipity cost for the three algorithms, lower values mean better performance

5.2.3 Discussion

First of all, we analyze the performances of the algorithms in term of total cost. As expected, the *normal mode* algorithm is the one which performs better, having a total cost of 0.46 ;the *surprise me!* mentor mode performance are lower (total cost 1.03) and finally the performance of the *surprise me!* random mode is very much lower: on average, 2.77 bad recommendations are proposed for each good (liked) recommendation.

These first results were expected: since the total cost does not take into consideration serendipity, a normal recommender system can take advantage of the fact that it proposes just items similar to the ones already liked by the user; a recommender system which tries to increase serendipity should take some risk, proposing music which is not necessarily liked by the user, and this can affect the general accuracy.

If we move our attention on serendipity, it is important to discuss the figures about the unexpected recommendations. The normal mode provides just 3 unexpected recommendations (out of 1,188); while the value should, theoretically, be 0, it can happen that a few recommendations are considered unexpected: if a playlist is created when the user has not provided any Facebook like yet but the user provides one or more likes through Mentor.FM during the execution of the playlist, this/those can determine a Facebook like (see 5.2.1 for details), that, however, affect the choice of artists only starting from the following playlist.

The *Surprise me!* Random mode does not provide any expected recommendation (even if, theoretically, it could) and finally for the *Surprise me!* Mentor mode most of the recommendations are unexpected (171) but there are some expected recommendation as well (56). The reason why both the *Surprise me!* algorithms could provide expected recommendations (and the mentor actually did) is explained in the following:

- for *random* profile users, the artists are extracted randomly, that means that, even if it is unlikely, they can be artists already liked by the user, or very similar to them;
- for *mentor* profile users, the artist belonging to the target musical world are probably⁵⁷ unexpected for the user, but the other artists who compose the playlist can represent, theoretically, expected recommendations.

Looking at the serendipity cost, we can conclude that the *Surprise me!* mentor algorithm is the one that performs better: the *Surprise me!* random algorithm has a serendipity cost which is more than 62.5% higher than the one of the competitor.

5.3 Evaluation of the cohesion

5.3.1 Metrics

To compare the performance of the techniques tested according to the cohesion, we proceeded following these steps:

1. For each user we considered all the songs listened during a defined experiment period, in the order the radio played them, and we measured the average similarity between one song and

⁵⁷ We stated “probably” because it is not guaranteed; as we explained, a user can like up to two artists of a MW and the MW will still be considered unknown (and therefore chosen)

the previous one, using as a similarity measure the *tagsim* measure (see 5.1.1).

2. We computed the average among all the average values found in the first step.

We called this measure Avg similarity (cohesion). It is important to highlight that this measure does not take into consideration just the cohesion of a playlists, taken by itself, but the cohesion of the whole listening experience of the user in a real-world scenario, in which the user can, for example, decide to stop listening the radio or load a new playlist before the current playlist comes to its end.

5.3.2 Results

In table 14 and 15 we show the results in term of the Avg similarity (cohesion) measure.

	Normal, random	Normal, DJ	Surprise me! Random	Surprise me! Mentor
Avg similarity (cohesion)	15.86%	35.30%	1.24%	28.92%

Table 14: Avg similarity (cohesion) index for Surprise me! Random and Mentor. Higher values mean better performance.

Since, during the period of the experiment, just 7 of the active users had a *random* profile for the normal mode⁵⁸, we also computed the measures taking the data from a period of about 90 hours immediately before (21 users having DJ profile and 22 having random). The results are presented in table 15.

⁵⁸ When we started this experiment, since we focused on the Surprise modes comparison, we set by default all the new users to DJ profile and during those days hundreds of new users subscribed; this explain the number of users having a DJ profile

	Normal mode, random	Normal mode, DJ
Avg similarity (cohesion)	19.11%	32.40%

Table 15: Avg similarity (cohesion) index for Normal mode random and DJ in a different period. Higher values mean better performance.

We also measured the impact on computation time of the DJ ordering technique, producing 10 playlist for the same users and tracking the time needed for the playlist creation script to be executed (t_{total}) and the time needed for the ordering part to be executed (t_{ord}). The average values for t_{total} and t_{ord} were 4.65879507 and 0.212055087 second.

5.3.3 Discussion

As expected, the *Surprise me!* mode random provides very low performance in term of cohesion: on average, the similarity of an artist to the previous one in the playlist was 1.24; since each song belonging to a playlist is chosen randomly, it is not very likely to choose a song whose artist is similar to the previous one in the playlist.

All the other modes provide much better performance; from the results, comparing the normal mode random with the normal mode DJ, we can see that applying the ordering method we proposed (see paragraphs 3.6.2 and 4.4.2) the performance can increase by 69.54%, with a limited impact on computation time: the execution time of the code which ordered the playlist represents the 4.55% of the total execution time. The *Surprise me!* mentor is, with an Avg similarity (cohesion) value of 28.92, the one which performs better after the normal mode DJ; this result is, again, not surprising since choosing the artists according to a path in the artists graph G_2 (see 3.6.3.2) - as the *Surprise me!* Mentor mode does for most of the them - guarantees a minimum similarity between an artist and the following one.

6

Conclusions, limitations and further research

The work presented in this thesis belongs to the research area of recommender systems, which is part of the wider research area of information filtering. This thesis, in particular, aims to propose techniques which can be adopted by recommender systems designers in order to increase serendipity while keeping an acceptable level of precision of the recommendations.

The research work has been conducted in the domain of music, which is a particularly suitable context in which to experiment serendipitous recommendations, mainly because of the relatively low cost of “bad” recommendations.

6.1 Contributions

The main contributions of our research work are summarized as follows.

1. A graph-based approach used to create a playlist of songs, each song played by a different artist, having in input the first and the last artist of the playlist. The method provides a smooth transition between each artist and the following one (in term of similarity between the two) using just collaborative filtering techniques and usage logs as inputs, no content analysis is required.

2. An application and validation of the affinity propagation clustering algorithm to the artist classification problem, again using just collaborative filtering techniques and usage logs as inputs, together with a folksonomy-based technique to synthetically represent each category of artist.
3. A recommendation technique - based on a variation of the classic user-based approach - which exploits the knowledge of the most eclectic users (we call them “mentors”), tries to find the clusters of music which are most likely to contain serendipitous items and propose them to the user trying to create a gradual path (playlist), using the technique described as the first contribution. In particular the playlist starts with songs by artists who tend to be borderline in respect to the user's taste and continues with songs by artists who tend to be, gradually, closer to the most representative artist of the proposed cluster.
4. A novel serendipity measure for recommender systems, called serendipity cost, which takes into consideration not only the number of serendipitous items proposed by a recommender system but also the cost in term of number of bad (disliked) recommendations produced.

An extensive work of implementation has been carried on in order to develop a real recommendation radio which has been used to test the techniques proposed, with the contribution of hundreds of users and for a period of several weeks, allowing the validation in a real environment. Since in our mentor approach time is an important factor and the order of the items in a playlist does matter, it would have been extremely hard to test it on static data without implementing a real recommendation radio. In particular, through the Mentor.FM radio, we compared our “Surprise me!” mentor algorithm to an approach which relies on randomness; our experiment suggested that the first performs better, considering the serendipity cost as

an evaluation parameter and a top-10 item-based recommender system as a baseline to detect which recommendations are supposed to be unexpected; the performance in term of cohesion was also better for the “Surprise me!” mentor. Furthermore, the “total cost” (measured as the total number of disliked songs over total number of liked songs) of the “Surprise me!” mentor approach, which can be considered an index of precision, is much lower than the cost related to the random approach and closer to the cost of a traditional top-10 item-based recommender systems (1.03 for the method proposed, 0.46 for the traditional recommender, 2.77 for the random)

From the validation point of view, we also highlight here that we compared affinity propagation clustering with a more traditional approach based on k-means, and our experiment showed not only that in a music classification problem the first performs better than the second, but also that measuring similarities between artists using a boolean approach gives results, in term of classification, comparable to the ones obtained using a Pearson Correlation approach, which is, considering the different computational effort, an interesting result.

6.2 Limitations and future research

As explained in details in paragraph 3.6.3.4 , our playlist creation approach sometimes creates a path which is optimal from the point of view of the length (it is the shortest one), provides smooth transitions between each artist in the playlist and the following one, but at some point goes through a path that moves too far from the destination. We are planning to improve our approach in order to overcome this limitation, in particular we would like to work on an algorithm which can relax the constraint of the distance between an artist and the following one in the playlist, if needed, and which accepts in input the level of “playlist smoothness” required.

Another aspect to improve is the diversity of the recommendation: choosing the shortest path between one artist and another leads to always propose the same intermediate artists: being able to produce several alternative paths, even if suboptimal, could increase diversity.

One limitation we should highlight about the experiment is that we cannot be completely sure about which recommendations are expected and which are not: users could like an artist without having “liked” it on Facebook or the artist could be not recognized by our system because is out of our dataset_30000 dataset or due a slightly different artist name used on Facebook. However, we think the “Facebook likes” recognized by our system can represent in many cases a good approximation of the music profile of the users. The only alternative would be directly asking to users which are the unexpected suggestions; even in this case, however, since you cannot be sure that all users always tell the system if a songs suggestion is unexpected, there could be false negative unexpected recommendations.

From the validation point of view, it would be interesting to compare our “Surprise me!” mentor algorithm also with [Oku, K. and Hattori, F. 2011], [Abassi et al., 2009] and with a traditional user-based collaborative filtering algorithm, providing an implementation of the three algorithms for Mentor.FM.

More on the implementation side, we would like to overcome the limitation of the string-based approach we currently use to match the artists the user liked on Facebook, the artists belonging to our dataset (coming from Last.FM) and the artists who are in the archive of our media partner (7digital); since the name of the artist could be written in a slightly different way in different systems and since there are also cases of homonymy, an approach based on a unique identification number would work better. Not only this could increase the user satisfaction but could also increase the precision of the method we use to recognize the artists the user is already familiar with, that could be considered, as we explain in details in 5.2.3 , a

limitation of our experiment by itself. We are looking forward to the evolution of vocabularies of music and in particular to the Project Rosetta Stone by The Echo Nest⁵⁹, which should allow the translation between multiple ID spaces; at the moment it seems that Last.FM is not supported and that Facebook support is in beta.

⁵⁹ <http://developer.echonest.com/docs/v4/#project-rosetta-stone>

References

- Abbassi, Z., Amer-Yahia, S., Lakshmanan, L. V., Vassilvitskii, S. and Yu, S. 2009. Getting recommender systems to think outside the box. In *Proceedings of the 3rd ACM conference on Recommender systems (New York, NY, USA, October 22-25, 2009)*.
- Adomavicius, G. and 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*, vol. 17, no. 6, June 2005.
- Alodhaibi, K., Brodsky, A. and Mihaila, G.A. 2011. A confidence-based recommender with adaptive diversity. *Proceedings of the IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MDCM)*.
- Barrington, L., Oda, R. and Lanckriet, G. 2009. Smarter than genius? Human evaluation of music recommender systems. In *Proceedings of the 10th International Conference on Music Information Retrieval (Kobe, Japan October 26-30, 2009)*.
- Bischoff, K., S. Firan, C., Nejd, W. and Paiu, R. 2008. Can all tags be used for search? In *Proceeding of the 17th ACM conference on Information and knowledge management (Napa Valley, CA, USA, October 26 - 30, 2008)*.

References

- Breese, J.S., Heckerman, D., and Kadie, C. 1998. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (Minneapolis, Minnesota, USA, July 1998)*.
- Burke, R. 2007. Hybrid Web Recommender Systems. In *Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.). The Adaptive Web: Methods and Strategies of Web Personalization., Lecture Notes in Computer Science, Vol. 4321, pp. 377-408, Springer, Berlin-Heidelberg, 2007.*
- Candillier, L., Meyer, F. and Boullé, M. 2007. Comparing state-of-the-art collaborative filtering systems. In *Petra Perner, editor; 5th International Conference on Machine Learning and Data Mining in Pattern Recognition; MLDM'2007, Leipzig, Germany, 18-20 July 2007; Lecture Notes in Computer Science, LNAI 4571, pages 548-562.*
- Celma, O. 2008. Music Recommendation and Discovery in the Long Tail. *Ph.D. thesis.*
- Dueck, D., Frey, B. J., Jojic, N., Jojic, V., Gjaever, G., Emili, A., et al. 2008. Constructing treatment portfolios using affinity propagation. *Lecture Notes in Computer Science, 2008, Volume 4955/2008, pp. 360-371. Springer.*
- Fields,, B. 2011. Contextualize Your Listening: The Playlist as Recommendation Engine (Ph.D. Thesis).
- Flexer, A., Schnitzer, D., Gasser, M., and Widmer, G. 2008. Playlist Generation Using Start and End Songs. *Ninth International Conference on Music Information Retrieval.*
- Flickr: Discussing Machine tags in Flickr API, [online], <http://www.flickr.com/groups/api/discuss/72157594497877875/>
- Floyd, R. W. 1962. Algorithm 97: shortest path, *Comm. ACM (1962) 345.*

References

- Frey, B.J. And Dueck, D. 2007. Clustering by passing messages between data points. *Science*, vol. 315 pp. 972–976, 2007.
- Ge, M., Delgado-Battenfeld, C. and Jannach, D. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems (RecSys '10)*. ACM, New York, NY, USA, 257-260.
- Goldberg, K., T. Roeder, D. Gupta, and C. Perkins, 2001. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval Journal*, Volume 4, Number 2, pp. 133-151. Springer.
- Golder, S. A. and Huberman, B. A. 2005. The Structure of Collaborative Tagging Systems, *HPL Technical Report*.
- Gruber, T. 2007. Ontology of folksonomy: A mash-up of apples and oranges, *International Journal on Semantic Web & Information Systems* 3(2).
- Halpin, H., Robu, V. and Shepherd, H. 2007. The complex dynamics of collaborative tagging. In *Proceedings of the 16th international conference on World Wide Web (WWW '07)*. ACM, New York, NY, USA.
- Hepp, M., Bachlechner, D. and Siorpaes K. 2006. OntoWiki: community-driven ontology engineering and ontology usage based on Wikis, *Proceedings of the 2006 international symposium on Wikis, Odense, Denmark, 2006 Pages: 143 – 144*
- Hijikata, Y. et al. Discovery-oriented collaborative filtering for improving user satisfaction. In *proceedings of the 13th international Conference on intelligent User interfaces (Sanibel Island, Florida, USA, 2009)*.
- Herlocker, J. L., Konstan, J. A., Terveen, L. G. and Riedl, J. T. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22, 1 (January 2004), 5-53.

References

- Iaquinta, L. et al. Recommendations toward Serendipitous Diversions. In *proceedings of the 9th international Conference on intelligent Systems Design and Applications (Washington, 2009)*.
- Jia, Y., Wang, J, Zhang, C. and Hua ,X. 2008. Finding image exemplars using fast sparse affinity propagation. In *Proceeding of the 16th ACM international conference on Multimedia (Vancouver, British Columbia, Canada, October 27-November 1, 2008)*.
- Knees, P., Pampalk, E. and Widmer., .. 2004. Artist classification with web-based data. In *Proceedings of the 5th International Conference on Music Information Retrieval (Barcelona, Spain October 10-14, 2004)*.
- Knees, P., Pohle, T., Schedl, M., and Widmer, G. 2006. Combining audio-based similarity with web-based data to accelerate automatic music playlist generation. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval (MIR '06)*. ACM, New York, NY, USA.
- Lamere, P. 2011. Blog post, <http://musicmachinery.com/2011/10/23/what-is-so-special-about-music/>
- Mathes, A. 2004. Folksonomies – Cooperative Classification and Communication Through Shared Metadata, [online], <http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html>
- McNee, S. M., Riedl, J. and Konstan, J. A. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI '06 extended abstracts on Human factors in computing systems (CHI EA '06)*. ACM, New York, NY, USA.
- Munson, S. A. and Resnick, P. 2010. Presenting diverse political opinions: how and how much. In *Proceedings of the 28th international conference on Human factors in computing systems (CHI '10)*. ACM, New York, NY, USA, 1457-1466.

References

- Murakami, T., Mori, K. and Orihara, R. Metrics for evaluating the serendipity of recommendation lists. In *Ken Satoh, Akihiro Inokuchi, Katashi Nagao, and Takahiro Kawamura, editors, JSAI, volume 4914 of Lecture Notes in Computer Science, pages 40–46. Springer, 2008*
- Neil Hurley and Mi Zhang. 2011. Novelty and Diversity in Top-N Recommendation -- Analysis and Evaluation. *ACM Trans. Internet Technol.* 10, 4, Article 14 (March 2011), 30 pages.
- Onuma, K., Tong, H. and Faloutsos, C. 2009. TANGENT: a novel, 'Surprise me', recommendation algorithm. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '09). ACM, New York, NY, USA, 657-666.*
- Oku, K. and Hattori, F. 2011. Fusion-based Recommender System for Improving Serendipity. *Proceedings of the Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011), at the 5th ACM International Conference on Recommender Systems (RecSys 2011). Chicago, Illinois, USA, 23 October 2011.*
- Passant, A. and Laublet, P. 2008. Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data, *Linked Data on The Web (LDOW2008), Workshop, WWW2008, Pékin, April.*
- Quintarelli, E. 2005. Folksonomies: power to the people, [online], paper presented at the *ISKO Italy-UniMIB meeting*, <http://www.iskoi.org/doc/folksonomies.htm>.
- Ricci, F., Rokach, L., and Shapira, B. 2010. Recommender Systems Handbook. Springer.
- Sarwar, B. M., Karypis, G., Konstan, J. A. and Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International World Wide Web Conference (Hong Kong, May 1-5, 2001).*

References

- Schafer, J.B., Frankowski, D., Herlocker, J., and Sen, S. 2007. Collaborative Filtering Recommender Systems, appears in *The Adaptive Web*, Springer Verlag.
- Shavitt, Y. and Weinsberg, U. 2009. Song clustering using peer-to-peer co-occurrences. In *Proceedings of the 11th IEEE International Symposium on Multimedia (San Diego, California, USA December 14-16, 2009)*.
- Sordo, M., Celma, O., Blech, M., and Gaus, E. The quest for musical genres: Do the experts and the wisdom of crowds agree? In *Proc. of ISMIR, Philadelphia, PA, USA, pp. 255–260, 2008*.
- Schein, A. I., Popescul, A., Ungar, L. H. and Pennock, D. M. Methods and metrics for cold-start recommendations. In *Proc. of the 25th Annual Intl. ACM SIGIR Conf., 2002*.
- Schwartz, B. 2004. The Paradox of Choice. *ECCO, New York*.
- Slaney, M. and White, W. 2011. Recommendation diversity. *Patent 7860862*.
- Swearingen, K. & Sinha, R. 2001. Beyond Algorithms: An HCI Perspective on Recommender Systems, in *ACM SIGIR Workshop on Recommender Systems*.
- Tacchini, E. and Damiani, E. 2011. Can a Rock song have a Jazz audience? Relationship between folksonomy and collaborative filtering in music recommender systems. In *Proceedings of the 3rd ACM International Conference on Web Science (Koblenz, Germany, June 14-17 2011)*.
- Toms, E. G. 1999. Serendipitous Information Retrieval. *Library and Information Science, 11–12*.
- Wang, D., Li, T. and Ogihara, M. 2010. Are tags better than audio features? The effect of joint use of tags and audio content features for artistic style

References

clustering. In *Proceedings of the 11th International Conference on Music Information Retrieval (Utrechts, Netherlands August 9-13, 2010)*.

Xie, X. and Wu, F. 2008. Automatic Video Summarization by Affinity Propagation Clustering and Semantic Content Mining. In *Proceedings of the International Symposium on Electronic Commerce and Security (Guangzhou, China , August 3-5, 2008)*.

Zeng, W., Shang, M.-S., Zhang, Q.-M., Lü, L. and Zhou, T. 2010. Can dissimilar users contribute to accuracy and diversity of personalized recommendation? *Int. J. Mod. Phys.*

Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R. and Zhang, Y.-C. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems, *Proc. Natl. Acad. Sci. U.S.A.* 107 (2010) 4511.

A

Publications, presentations and awards

Publications and presentations

2008

- Presentation: “Toward trust-aware folksonomy evolution for blogosphere learning” (author of the work: Eugenio Tacchini), Advanced International Summer School “e-Business and Complexity: New Management Practices” 2008 Session: Managing Uncertainty and Competition Through Dynamic Learning Networks, Ostuni, Italy.

2009

- Paper presentation and publication “Experiments with wikipedia cross-language data fusion” (Authors: Eugenio Tacchini, Andreas Schultz, Christian Bizer), 5th Workshop on Scripting and Development for the Semantic Web (SFSW2009), 6th European Semantic Web Conference, Heraklion, Greece, 31 May.
- Presentation “Web data fusion in the Wikipedia/DBpedia domain”, IFIP 2.6 - 2nd Research Workshop 2009, Querying the Data Web, August 28th 2009, Lyon, France.

2010

- Presentation, “Handling eclectic tastes in recommender systems: novelty, serendipity and mentors”, 4th Workshop of the Multimedia, Distributed and Pervasive Systems doctoral college, University of Passau, Germany, June 14th-18th 2010.
- Presentation, “Serendipitous mentoring in recommender systems”, 5th Workshop of the Multimedia, Distributed and Pervasive Systems doctoral college, INSA de Lyon, Germany, December 13th-17th 2010.

2011

- Poster publication. Tacchini, E. and Damiani, E. 2011. Can a Rock song have a Jazz audience? Relationship between folksonomy and collaborative filtering in music recommender systems. In Proceedings of the 3rd ACM International Conference on Web Science (Koblenz, Germany, June 14-17 2011).
- Paper presentation and publication. Tacchini, E. and Damiani, E. 2011. What is a Musical World? An affinity propagation approach. In Proceedings of the 1st International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies (MIRUM) (Scottsdale, AZ, Nov 28-Dec 1 2011).

Awards

2010

- Winner, with the project Mentorex, of the Working Capital contest for best research ideas held by Telecom Italia S.p.a., the largest Italian telecommunications company.

B

36 Musical Worlds according to Boolean AP

For each musical world derived from dataset_3000, the representative artist and the list of all the artists included are shown. For each artist, we also show the most important tag of the relative tag cloud.

Musical World 1

Representative artist: radiohead - alternative

List: radiohead - alternative, the beatles - classic rock, coldplay - rock, red hot chili peppers - rock, muse - alternative rock, pink floyd - Progressive rock, the killers - indie, system of a down - metal, daft punk - electronic, the cure - new wave, placebo - alternative, depeche mode - electronic, nirvana - Grunge, arctic monkeys - indie, foo fighters - rock, nine inch nails - industrial, sigur rós - post-rock, massive attack - trip-hop, amy winehouse - soul, portishead - trip-hop, björk - electronic, bloc party - indie, kings of leon - indie, the white stripes - rock, oasis - britpop, the prodigy - electronic, the smashing pumpkins - alternative rock, jack johnson - acoustic, franz ferdinand - indie, air - electronic, moby - electronic, gorillaz - alternative, incubus - rock, r.e.m. - rock, beck - alternative, the smiths - indie, the strokes - rock, tool - Progressive metal, interpol - indie, snow patrol - indie, pearl jam - Grunge, queens of the stone age - Stoner Rock, røyksopp -

electronic, the kooks - indie, pixies - alternative, rage against the machine - rock, marilyn manson - industrial, the chemical brothers - electronic, joy division - post-punk, mgmt - electronic, beastie boys - Hip-Hop, norah jones - jazz, blur - britpop, jamiroquai - funk, keane - britpop, a perfect circle - alternative rock, manu chao - reggae, goldfrapp - electronic, the cranberries - rock, mogwai - post-rock, tori amos - female vocalists, pj harvey - female vocalists, nick cave and the bad seeds - rock, fatboy slim - electronic, morrissey - indie, travis - britpop, the mars volta - Progressive rock, new order - new wave, garbage - rock, cake - alternative, soundtrack - Soundtrack, the cardigans - pop, gnarls barkley - funk, jeff buckley - singer-songwriter, nouvelle vague - Bossa Nova, dave matthews band - rock, counting crows - rock, stereophonics - rock, the verve - britpop, weezer - rock, the streets - Hip-Hop, soundgarden - Grunge, manic street preachers - rock, mew - indie, the dresden dolls - punk cabaret, moby - electronic, ben harper - singer-songwriter, pulp - britpop, black rebel motorcycle club - rock, thom yorke - electronic, eddie vedder - singer-songwriter, the dandy warhols - indie, elbow - indie, john frusciante - alternative, archive - trip-hop, doves - indie, morphine - jazz, suede - britpop, emilie simon - french, deus - alternative, mazy star - dream pop, james - britpop, the john butler trio - acoustic, madrugada - rock, the dø - indie, kashmir - danish, the seatbelts - jazz, k's choice - rock, ride - shoegaze, afterhours - italian, amanda palmer - punk cabaret, subsonica - italian, alina orlova - female vocalists, grinderman - Garage Rock

Musical World 2

Representative artist: the rolling stones - classic rock

List: queen - classic rock, u2 - rock, led zeppelin - classic rock, david bowie - rock, bob dylan - folk, the rolling stones - classic rock, johnny cash -

Appendix B: 36 Musical Worlds according to Boolean AP

country, the doors - classic rock, tom waits - blues, bob marley - reggae, the clash - punk, jimi hendrix - classic rock, the who - classic rock, elvis presley - rock n roll, bob marley & the wailers - reggae, aerosmith - rock, frank sinatra - jazz, bruce springsteen - rock, the beach boys - classic rock, neil young - classic rock, the velvet underground - rock, simon & garfunkel - folk, dire straits - classic rock, creedence clearwater revival - classic rock, eric clapton - classic rock, the kinks - classic rock, nina simone - jazz, the police - rock, leonard cohen - singer-songwriter, elton john - pop, sting - rock, nirvana - Grunge, frank zappa - Progressive rock, lenny kravitz - rock, fleetwood mac - classic rock, genesis - Progressive rock, john lennon - classic rock, jethro tull - Progressive rock, the jimi hendrix experience - classic rock, ray charles - jazz, eagles - classic rock, santana - rock, van morrison - classic rock, billy joel - classic rock, phil collins - pop, blondie - new wave, the raconteurs - rock, peter gabriel - Progressive rock, cat stevens - folk, janis joplin - classic rock, the black keys - blues rock, jefferson airplane - Psychedelic Rock, lou reed - rock, duran duran - 80s, electric light orchestra - classic rock, lynyrd skynyrd - classic rock, stone temple pilots - Grunge, james brown - funk, cream - classic rock, elvis costello - rock, tracy chapman - female vocalists, buena vista social club - latin, paul simon - singer-songwriter, eagles of death metal - rock, supertramp - classic rock, iggy pop - punk, the stone roses - britpop, serge gainsbourg - french, paul mccartney - classic rock, otis redding - soul, t. rex - glam rock, grateful dead - classic rock, bee gees - Disco, sheryl crow - female vocalists, steely dan - classic rock, inxs - rock, primal scream - rock, b.b. king - blues, the byrds - classic rock, supergrass - britpop, echo & the bunnymen - post-punk, mark knopfler - rock, tom petty and the heartbreakers - classic rock, the jam - punk, chuck berry - Rock and Roll, george harrison - classic rock, john lee hooker - blues, the stooges - punk, the band - classic rock, patti smith - rock, electric six - rock, the allman brothers band - Southern Rock, rod stewart - classic rock, eurythmics - 80s, muddy waters - blues, roxy music - glam

Appendix B: 36 Musical Worlds according to Boolean AP

rock, crowded house - rock, violent femmes - alternative, sam cooke - soul, the blues brothers - blues, donovan - folk, david gilmour - Progressive rock, carpenters - pop, mano negra - ska, stevie ray vaughan and double trouble - blues, syd barrett - psychedelic, stevie ray vaughan - blues, dusty springfield - soul, the replacements - alternative, joe cocker - classic rock, buddy holly - rockabilly, roy orbison - classic rock, the animals - classic rock, chicago - classic rock, rodrigo y gabriela - acoustic, blind melon - Grunge, the mamas & the papas - classic rock, the b-52's - new wave, the yardbirds - classic rock, chris isaak - rock, love - psychedelic, phish - Jam, paul weller - rock, hank williams - country, the black crowes - rock, fabrizio de andré - italian, yes - Progressive rock, nancy sinatra - female vocalists, the stranglers - new wave, neil diamond - singer-songwriter, tom petty - classic rock, roger waters - Progressive rock, j.j. cale - blues, "weird al" yankovic - comedy, chris rea - rock, jane's addiction - alternative rock, elvis costello & the attractions - new wave, lucinda williams - Alt-country, townes van zandt - folk, howlin' wolf - blues, monty python - comedy, wolfmother - rock, the moody blues - classic rock, joan baez - folk, robert johnson - blues, franco battiato - italian, steve miller band - classic rock, drive-by truckers - Alt-country, america - classic rock, jeff beck - classic rock, the sonics - Garage Rock, buddy guy - blues, nick cave - rock, hall & oates - 80s, the monkees - 60s, big star - power pop, tom jones - pop, the cars - new wave, steve earle - Alt-country, kula shaker - britpop, the hollies - 60s, robert plant & alison krauss - folk, jackson browne - singer-songwriter, the pretenders - rock, jerry lee lewis - rockabilly, the wallflowers - rock, carole king - female vocalists, vasco rossi - italian, elio e le storie tese - italian, rory gallagher - blues rock, john denver - country, the doobie brothers - classic rock, steppenwolf - classic rock, cowboy junkies - Alt-country, neil young & crazy horse - classic rock, ten years after - classic rock, buffalo springfield - classic rock, meat puppets - Grunge, lucio battisti - italian, the ventures - Surf, paul mccartney & wings - classic rock, fairport convention - folk, bløf

- dutch, willie nelson - country, john mellencamp - rock, bryan ferry - 80s, bobby darin - jazz, little richard - Rock and Roll, midnight oil - rock, the jon spencer blues explosion - rock, canned heat - blues, harry nilsson - singer-songwriter, john cale - singer-songwriter, the tragically hip - Canadian, booker t. & the mg's - soul, the supremes - soul, television - post-punk, joe jackson - new wave, emerson, lake & palmer - Progressive rock, tim buckley - singer-songwriter, traveling wilburys - classic rock, acda en de munnik - dutch, leadbelly - blues, francesco de gregori - italian

Musical World 3

Representative artist: sufjan stevens - indie

List: death cab for cutie - indie, arcade fire - indie, belle and sebastian - indie, modest mouse - indie, sufjan stevens - indie, the shins - indie, eliott smith - singer-songwriter, feist - female vocalists, beirut - folk, cat power - indie, regina spektor - female vocalists, bright eyes - indie, iron & wine - folk, damien rice - singer-songwriter, john mayer - singer-songwriter, the decemberists - indie, the postal service - indie, kings of convenience - indie, broken social scene - indie, explosions in the sky - post-rock, josé gonzález - acoustic, nick drake - folk, tegan and sara - indie, the flaming lips - indie, eels - indie, the national - indie, andrew bird - indie, imogen heap - female vocalists, spoon - indie, architecture in helsinki - indie, stars - indie, band of horses - indie, bon iver - folk, yann tiersen - instrumental, the magnetic fields - indie, antony and the johnsons - singer-songwriter, metric - indie, rilo kiley - indie, ryan adams - singer-songwriter, kate bush - female vocalists, fiona apple - female vocalists, rufus wainwright - singer-songwriter, calexico - indie, ben folds - singer-songwriter, phoenix - indie, the new pornographers - indie, the mountain goats - indie, emiliana torrini - female vocalists, patrick wolf - indie, clap your hands say yeah - indie,

Appendix B: 36 Musical Worlds according to Boolean AP

okkervil river - indie, wilco - Alt-country, nada surf - indie, minus the bear - indie, jens lekman - swedish, joni mitchell - folk, the album leaf - post-rock, they might be giants - alternative, frou frou - indie, aimee mann - female vocalists, flight of the conchords - comedy, m. ward - folk, my morning jacket - indie, barenaked ladies - rock, cold war kids - indie, the notwist - indie, ani difranco - folk, camera obscura - indie pop, david gray - singer-songwriter, the go! team - indie, badly drawn boy - indie, kimya dawson - indie, neko case - Alt-country, ben folds five - alternative, feist - female vocalists, ray lamontagne - singer-songwriter, pinback - indie, cursive - indie, mirah - indie, final fantasy - indie, camille - french, the hold steady - indie rock, silversun pickups - indie rock, tindersticks - indie, the weakerthans - indie, suzanne vega - female vocalists, devotchka - Gypsy, granddaddy - indie, ingrid michaelson - female vocalists, mates of state - indie, she & him - folk, aqualung - indie, the weepies - folk, mewithoutyou - post-hardcore, the bird and the bee - indie, ben kweller - indie, rachael yamagata - female vocalists, guster - indie, gomez - indie, ted leo and the pharmacists - indie, beth orton - female vocalists, the appleseed cast - indie, midlake - indie, sondre lerche - singer-songwriter, sparklehorse - indie, fountains of wayne - rock, eisley - indie, amadou & mariam - african, tilly and the wall - indie, lambchop - Alt-country, glen hansard and markéta irglová - folk, kaki king - acoustic, rogue wave - indie, the american analog set - indie, original broadway cast - Broadway, the divine comedy - indie, dispatch - indie, josh ritter - singer-songwriter, pedro the lion - indie, azure ray - indie, teenage fanclub - indie, psapp - electronic, jon brion - Soundtrack, great lake swimmers - folk, sun kil moon - folk, matt costa - singer-songwriter, tunng - folktronica, mitch hedberg - comedy, voxtrot - indie, liz phair - female vocalists, amiina - icelandic, frightened rabbit - Scottish, keren ann - female vocalists, brandi carlile - female vocalists, i am kloom - indie, ryan adams & the cardinals - Alt-country, conor oberst - singer-songwriter, the frames - irish, nickel creek - bluegrass, bishop allen -

indie, the avett brothers - folk, charlotte gainsbourg - french, martha wainwright - singer-songwriter, angus & julia stone - folk, ben lee - indie, laura veirs - singer-songwriter, songs: ohia - singer-songwriter, cloud cult - indie, gillian welch - Alt-country, owen - indie, the polyphonic spree - indie, alela diane - folk, my brightest diamond - female vocalists, ben harper - singer-songwriter, rocky votolato - singer-songwriter, g. love & special sauce - blues, red house painters - slowcore, the wedding present - indie, indigo girls - folk, pete yorn - singer-songwriter, seabear - icelandic, someone still loves you boris yeltsin - indie, isobel campbell & mark lanegan - singer-songwriter, shearwater - folk, uncle tupelo - Alt-country, joseph arthur - singer-songwriter, jenny lewis with the watson twins - indie, old crow medicine show - bluegrass, emily haines & the soft skeleton - female vocalists, soul coughing - alternative, beulah - indie, afghan whigs - alternative rock, matt pond pa - indie, damien jurado - singer-songwriter, spinvis - dutch, richard hawley - singer-songwriter, patrick watson - indie, margot & the nuclear so and so's - indie, billy bragg & wilco - Alt-country, the dears - indie, patty griffin - folk, ra ra riot - indie, her space holiday - indie, dave matthews - rock, the be good tanyas - folk, joan as police woman - singer-songwriter, gregory and the hawk - indie, pretty girls make graves - indie, micah p. hinson - singer-songwriter, david byrne - alternative, alexi murdoch - singer-songwriter

Musical World 4

Representative artist: rihanna - pop

List: madonna - pop, michael jackson - pop, britney spears - pop, rihanna - pop, [unknown] - mysterious, avril lavigne - pop, lily allen - pop, nelly furtado - pop, lady gaga - pop, justin timberlake - pop, maroon 5 - rock, katy perry - pop, beyoncé - rnb, jason mraz - singer-songwriter, abba - pop, alanis

Appendix B: 36 Musical Worlds according to Boolean AP

morissette - female vocalists, black eyed peas - Hip-Hop, robbie williams - pop, dido - pop, christina aguilera - pop, james blunt - pop, mika - pop, alicia keys - soul, kylie minogue - pop, timbaland - Hip-Hop, kelly clarkson - pop, the fray - alternative, the pussycat dolls - pop, the ting tings - indie, mariah carey - pop, akon - Hip-Hop, duffy - soul, david guetta - House, pink - pop, cascada - dance, no doubt - rock, katie melua - jazz, michael bublé - jazz, kt tunstall - female vocalists, amy macdonald - female vocalists, ne-yo - rnb, leona lewis - pop, sarah mclachlan - female vocalists, gwen stefani - pop, bryan adams - rock, t.a.t.u. - pop, roxette - pop, chris brown - rnb, scissor sisters - pop, basshunter - dance, céline dion - pop, enrique iglesias - pop, backstreet boys - pop, usher - rnb, fergie - pop, p!nk - pop, adele - soul, sugababes - pop, onerepublic - rock, george michael - pop, natasha bedingfield - pop, jennifer lopez - pop, taylor swift - country, colbie caillat - female vocalists, janet jackson - pop, hilary duff - pop, the veronicas - pop, james morrison - singer-songwriter, benny benassi - electronic, destinys child - r'n'b, miley cyrus - pop, groove coverage - dance, seal - soul, spice girls - pop, robyn - pop, sara bareilles - female vocalists, ashlee simpson - pop, maria mena - female vocalists, craig david - rnb, girls aloud - pop, savage garden - pop, bob sinclar - House, september - dance, missy elliott - Hip-Hop, jem - female vocalists, cyndi lauper - 80s, gavin degraw - singer-songwriter, ace of base - pop, natalie imbruglia - pop, a fine frenzy - female vocalists, ciara - rnb, dixie chicks - country, anastacia - pop, take that - pop, disney - Soundtrack, sophie ellis-bextor - pop, jordin sparks - pop,anouk - rock, kate ryan - dance, the script - pop rock, michelle branch - pop, shania twain - country, guru josh project - House, annie lennox - female vocalists, jewel - female vocalists, vanessa carlton - pop, gabriella cilmi - female vocalists, aqua - pop, josh groban - Classical, carrie underwood - country, cher - pop, *nync - pop, tina turner - pop, missy higgins - female vocalists, jay-z and linkin park - Hip-Hop, pmmp - finnish, jesse mccartney - pop, morandi - dance, lindsay lohan - pop, jojo - pop, sean kingston - rnb, mandy

moore - pop, anna nalick - female vocalists, aly & aj - pop, cassie - rnb, danity kane - pop, high school musical - Disney, ashanti - rnb, madcon - Hip-Hop, ashley tisdale - pop, donna summer - Disco, lionel richie - soul, kelly rowland - rnb, e-type - eurodance, daniel powter - pop, laurent wolf - House, whitney houston - pop, kat deluna - rnb, delta goodrem - pop, elisa - italian, milow - singer-songwriter, kerii hilson - rnb, leann rimes - country, booty luv - House, keith urban - country, david cook - rock, melanie c - pop, mario - rnb, jay sean - rnb, alex gaudino - House, david archuleta - pop, eiffel 65 - dance, vanessa hudgens - pop, hanson - pop, rob thomas - pop, blue - pop, jessica simpson - pop, infernal - dance, faith hill - country, eagle-eye cherry - rock, all saints - pop, luciano ligabue - italian, the saturdays - pop, paris hilton - pop, nicole scherzinger - rnb, darren hayes - pop, krezip - rock, fedde le grand - House, chris brown - rnb

Musical World 5

Representative artist: animal collective - experimental

List: sonic youth - alternative, animal collective - experimental, of montreal - indie, vampire weekend - indie, m.i.a. - Hip-Hop, fleet foxes - folk, devendra banhart - folk, tv on the radio - indie, cocorosie - indie, m83 - electronic, talking heads - new wave, neutral milk hotel - indie, ratatat - electronic, pavement - indie, blonde redhead - indie, girl talk - mashup, yeah yeah yeahs - indie, my bloody valentine - shoegaze, bonnie 'prince' billy - folk, múm - icelandic, fugazi - post-hardcore, joanna newsom - folk, bat for lashes - female vocalists, the kills - indie, stereolab - electronic, the jesus and mary chain - shoegaze, grizzly bear - indie, beck - alternative, peter bjorn and john - indie, dinosaur jr. - alternative, deerhoof - experimental, built to spill - indie rock, deerhunter - shoegaze, yo la tengo - indie, battles - math rock, low - slowcore, guided by voices - Lo-Fi, the zombies - 60s, the

Appendix B: 36 Musical Worlds according to Boolean AP

fall - post-punk, devo - new wave, captain beefheart & his magic band - experimental, the blood brothers - post-hardcore, the books - experimental, slowdive - shoegaze, the raveonettes - indie, the brian jonestown massacre - psychedelic, hüsker dü - punk, spiritualized - space rock, can - krautrock, minutemen - punk, broadcast - electronic, caribou - electronic, super furry animals - indie, the avalanches - electronic, au revoir simone - indie pop, eluvium - ambient, the microphones - Lo-Fi, the apples in stereo - indie, do make say think - post-rock, the breeders - alternative, gang of four - post-punk, beach house - dream pop, the moldy peaches - indie, the walkmen - indie, liars - experimental, the fiery furnaces - indie, sleater-kinney - riot grrrl, fever ray - electronic, black lips - Garage Rock, smog - singer-songwriter, portugal. the man - experimental, electrelane - indie, the unicorns - indie, daniel johnston - Lo-Fi, menomena - experimental, panda bear - experimental, black moth super rainbow - psychedelic, les savy fav - post-punk, the dodos - folk, silver jews - indie, casiotone for the painfully alone - Lo-Fi, vashti bunyan - folk, the blow - indie, islands - indie, no age - noise rock, asobi seksu - shoegaze, british sea power - indie, wire - post-punk, sebadoh - Lo-Fi, passion pit - electronic, scott walker - singer-songwriter, the thermals - indie, akron/family - folk, cornelius - electronic, why? - indie, the sea and cake - indie, butthole surfers - alternative, wolf parade - indie, efterklang - post-rock, destroyer - indie, arab strap - indie, sunset rubdown - indie, the beta band - indie, camera obscura - indie pop, ween - alternative, vetiver - folk, sparks - new wave, clinic - indie, dr. dog - indie, mercury rev - indie, the clientele - indie, frank black - alternative, jay reatard - Garage Punk, man man - experimental, dntel - electronic, herman düne - folk, arthur russell - Avant-Garde, coconut records - indie, moondog - Avant-Garde, st. vincent - female vocalists, gang gang dance - experimental, atlas sound - shoegaze, big black - noise rock, beat happening - Lo-Fi, john fahey - folk, cap'n jazz - emo, tapes 'n tapes - indie, xtc - new wave, lightning bolt - noise rock, department of eagles - indie, mount eerie - Lo-Fi,

blitzen trapper - indie, spacemen 3 - shoegaze, nico - female vocalists, the olivia tremor control - Elephant 6, american football - emo, robert wyatt - Progressive rock, brian eno & david byrne - experimental, the birthday party - post-punk, the go-betweens - indie pop, mission of burma - post-punk, dirty projectors - experimental, six organs of admittance - folk, matt & kim - indie, the pains of being pure at heart - shoegaze, marissa nadler - folk, grouper - ambient, rachel's - post-rock, the black angels - Psychedelic Rock, xiu xiu - experimental, q and not u - indie, boredom's - experimental, the dismemberment plan - indie, brian wilson - pop, dirty three - post-rock

Musical World 6

Representative artist: judas priest - heavy metal

List: metallica - thrash metal, ac/dc - hard rock, iron maiden - heavy metal, guns n' roses - hard rock, black sabbath - heavy metal, dream theater - Progressive metal, slayer - thrash metal, megadeth - thrash metal, deep purple - classic rock, bon jovi - rock, pantera - thrash metal, judas priest - heavy metal, alice in chains - Grunge, motörhead - heavy metal, scorpions - hard rock, kiss - hard rock, ozzy osbourne - heavy metal, sepultura - thrash metal, helloween - Power metal, rush - Progressive rock, van halen - hard rock, manowar - heavy metal, iced earth - Power metal, alice cooper - hard rock, rob zombie - industrial metal, death - death metal, hammerfall - Power metal, joe satriani - guitar virtuoso, mötley crüe - hard rock, machine head - thrash metal, stratovarius - Power metal, kreator - thrash metal, fear factory - industrial metal, def leppard - hard rock, testament - thrash metal, edguy - Power metal, dio - heavy metal, journey - classic rock, black label society - heavy metal, gamma ray - Power metal, toto - rock, rainbow - hard rock, velvet revolver - hard rock, anthrax - thrash metal, marillion - Progressive rock, thin lizzy - hard rock, buckethead - experimental, billy idol - rock,

queensrÿche - Progressive metal, skid row - hard rock, camel - Progressive rock, danzig - heavy metal, steve vai - guitar virtuoso, europe - hard rock, nevermore - Progressive metal, gary moore - blues, uriah heep - classic rock, blue Öyster cult - classic rock, bruce dickinson - heavy metal, accept - heavy metal, saxon - heavy metal, exodus - thrash metal, foreigner - classic rock, annihilator - thrash metal, zz top - classic rock, monster magnet - Stoner Rock, meat loaf - rock, boston - classic rock, king diamond - heavy metal, the darkness - rock, savatage - heavy metal, poison - hard rock, rage - heavy metal, whitesnake - hard rock, twisted sister - hard rock, kansas - classic rock, the cult - hard rock, running wild - heavy metal, overkill - thrash metal, heart - classic rock, grave digger - heavy metal, status quo - classic rock, styx - classic rock, w.a.s.p. - heavy metal, destruction - thrash metal, mr. big - hard rock, brainstorm - Power metal, cinderella - hard rock, hawkwind - space rock, gentle giant - Progressive rock, dream evil - Power metal, extreme - hard rock, airbourne - hard rock, nazareth - classic rock, bad company - classic rock, hanoi rocks - glam rock, primal fear - Power metal, gotthard - hard rock, hardcore superstar - hard rock, corrosion of conformity - Stoner Rock, spock's beard - Progressive rock, cheap trick - classic rock, axel rudi pell - heavy metal, yngwie malmsteen - guitar virtuoso, joe bonamassa - blues rock, eric johnson - guitar virtuoso, tesla - hard rock, merciful fate - heavy metal, dokken - hard rock, tankard - thrash metal, anthrax - thrash metal

Musical World 7

Representative artist: breaking benjamin - alternative rock

List: linkin park - rock, коЯn - Nu Metal, evanescence - rock, slipknot - metal, disturbed - metal, my chemical romance - emo, nickelback - rock, rise against - punk rock, sum 41 - punk rock, jimmy eat world - rock, 3

Appendix B: 36 Musical Worlds according to Boolean AP

doors down - rock, 30 seconds to mars - alternative rock, limp bizkit - Nu Metal, audioslave - rock, pendulum - Drum and bass, papa roach - rock, bullet for my valentine - metalcore, him - Love Metal, good charlotte - punk rock, breaking benjamin - alternative rock, three days grace - alternative rock, lostprophets - rock, avenged sevenfold - metalcore, goo goo dolls - rock, mindless self indulgence - industrial, guano apes - rock, lifehouse - rock, coheed and cambria - Progressive rock, atreyu - metalcore, static-x - industrial metal, staind - rock, godsmack - hard rock, p.o.d. - Nu Metal, hoobastank - rock, stone sour - metal, seether - rock, fort minor - Hip-Hop, creed - rock, the rasmus - rock, flyleaf - alternative rock, matchbox twenty - rock, mudvayne - Nu Metal, ill niño - Nu Metal, akira yamaoka - Soundtrack, live - rock, silverchair - Grunge, chevelle - alternative rock, bush - Grunge, drowning pool - Nu Metal, poets of the fall - rock, skillet - christian rock, chris cornell - rock, dope - industrial metal, sonic syndicate - Melodic Death Metal, alter bridge - rock, adema - Nu Metal, shinedown - rock, daughtry - rock, puddle of mudd - rock, hinder - rock, trapt - rock, soil - metal, celldweller - industrial, lumen - alternative, scars on broadway - alternative rock, sevendust - metal, collective soul - rock, our lady peace - rock, alien ant farm - rock, smash mouth - rock, cold - alternative rock, saliva - hard rock, blue october - alternative, finger eleven - alternative rock, crossfade - rock, noize mc - Hip-Hop, kid rock - rock, animal Джаз - alternative, flobots - Hip-Hop, cky - rock, the calling - rock, theory of a deadman - rock, fuel - rock, thousand foot krutch - christian rock, 12 stones - rock, 10 years - alternative rock, powerman 5000 - industrial metal, (hed) planet earth - rapcore, five for fighting - alternative, red - alternative rock, mushroomhead - Nu Metal, stephen lynch - comedy, coal chamber - Nu Metal, spineshank - Nu Metal, blindside - rock, taproot - Nu Metal, filter - rock, buckcherry - hard rock, vertical horizon - rock, demon hunter - metalcore, train - rock, nonpoint - Nu Metal, five finger death punch - metalcore, eve 6 - rock, crazy town - rapcore, jane air - alternative, black

stone cherry - hard rock, 7paca - alternative, tracktor bowling - alternative, chris daughtry - rock, dead poetic - post-hardcore.

Musical World 8

Representative artist: die Ärzte - punk rock

List: rammstein - industrial metal, tenacious d - rock, billy talent - punk rock, the hives - rock, die Ärzte - punk rock, mando diao - rock, bloodhound gang - rock, serj tankian - alternative metal, ska-p - ska, die toten hosen - punk rock, beatsteaks - rock, richard cheese - lounge, in extremo - folk metal, seeed - reggae, adam green - indie, deichkind - electronic, oomph! - industrial, peter fox - german, clueso - german, subway to sally - folk metal, die fantastischen vier - german, farin urlaub - german, tocotronic - indie, böhse onkelz - rock, fettes brot - german, the cat empire - ska, k.i.z. - rap, schandmaul - folk rock, sportfreunde stiller - german, sunrise avenue - rock, sido - rap, danko jones - rock, everlast - rock, bushido - rap, mia - german, madsen - german, kettcar - indie, 2raumwohnung - german, blumentopf - german, jan delay - reggae, asp - Gothic, culcha candela - reggae, tomte - indie, herbert grönemeyer - german, clawfinger - crossover, juli - german, silbermond - german, j.b.o. - fun metal, rosenstolz - german, samsas traum - Gothic, polarkreis 18 - electronic, the dubliners - irish, element of crime - german, juliette and the licks - rock, samy deluxe - hip hop, falco - 80s, heather nova - female vocalists, prinz pi - Deutschrap, annett louisian - german, letzte instanz - folk metal, absolute beginner - hip hop, wir sind helden - german, megaherz - industrial metal, dendemann - hip hop, freundeskreis - german, curse - hip hop, knorkator - fun metal, donots - punk rock, ich + ich - german, tanzwut - folk metal, helge schneider - comedy, dynamite deluxe - hip hop.

Musical World 9

Representative artist: rancid - punk

List: green day - punk rock, blink-182 - punk rock, the offspring - punk rock, ramones - punk, afi - punk, bad religion - punk, nofx - punk, misfits - horror punk, sublime - ska, dropkick murphys - punk, flogging molly - irish, rancid - punk, alkaline trio - punk, gogol bordello - gypsy punk, dead kennedys - punk, millencolin - punk rock, anti-flag - punk, against me! - punk, reel big fish - ska, less than jake - ska, sex pistols - punk, black flag - punk, pennywise - punk, the pogues - irish, bad brains - hardcore, 311 - alternative, descendents - punk, turbonegro - death punk, social distortion - punk, goldfinger - punk, buzzcocks - punk, me first and the gimme gimmes - punk, minor threat - hardcore, madness - ska, the distillers - punk, the specials - ska, the cramps - psychobilly, streetlight manifesto - ska, lagwagon - punk, mxpx - punk, mad caddies - ska, the gaslight anthem - punk rock, hole - Grunge, no use for a name - punk, the bouncing souls - punk, refused - hardcore, operation ivy - punk, propagandhi - punk, billy bragg - folk, the presidents of the united states of america - rock, agnostic front - hardcore, sick of it all - hardcore, hot water music - punk, tiger army - psychobilly, the skatalites - ska, the mighty mighty bosstones - ska, jawbreaker - punk, horrorpops - psychobilly, strike anywhere - melodic hardcore, everclear - rock, the exploited - punk, the damned - punk, suicidal tendencies - hardcore, catch 22 - ska, backyard babies - hard rock, stray cats - rockabilly, slightly stoopid - reggae, the adicts - punk, raised fist - hardcore, the aquabats - ska, a wilhelm scream - melodic hardcore, strung out - punk, bedouin soundclash - reggae, the donnas - rock, the lawrence arms - punk, h2o - hardcore, nekromantix - psychobilly, the living end - punk, big d and the kids table - ska, mudhoney - Grunge, chumbawamba - alternative, screeching weasel - punk, the brian setzer orchestra - swing, reverend horton heat - psychobilly, leftöver crack - punk, andrew w.k. -

rock, the casualties - punk, the toy dolls - punk, good riddance - punk, mad sin - psychobilly, crass - punk, bikini kill - riot grrrl, the suicide machines - punk, therapy? - rock, 7seconds - hardcore, lifetime - melodic hardcore, cock sparrer - Oi, circle jerks - punk, the meteors - psychobilly, oingo boingo - new wave, the bronx - hardcore, the vandals - punk, rx bandits - ska, stiff little fingers - punk, dillinger four - punk, the slackers - ska, mc chris - nerdcore, distemper - ska punk, the unseen - punk, gorilla biscuits - hardcore, lucero - Alt-country, bad manners - ska, bomb the music industry! - ska, great big sea - folk, unwritten law - rock.

Musical World 10

Representative artist: ensiferum - folk metal

List: in flames - Melodic Death Metal, opeth - Progressive metal, children of bodom - Melodic Death Metal, blind guardian - Power metal, amon amarth - Melodic Death Metal, arch enemy - Melodic Death Metal, dimmu borgir - black metal, cradle of filth - black metal, dark tranquillity - Melodic Death Metal, ensiferum - folk metal, soilwork - Melodic Death Metal, amorphis - Progressive metal, korpiklaani - folk metal, dragonforce - Power metal, finntroll - folk metal, soulfly - thrash metal, volbeat - heavy metal, sentenced - Gothic Metal, eluveitie - folk metal, pain - industrial metal, agalloch - doom metal, dethklok - death metal, norther - Melodic Death Metal, hypocrisy - death metal, deathstars - industrial metal, bloodbath - death metal, die apokalyptischen reiter - death metal, sabaton - Power metal, turisas - folk metal, kalmah - Melodic Death Metal, scar symmetry - Melodic Death Metal, machinae supremacy - sid metal, kataklysm - death metal, cmx - finnish, equilibrium - folk metal, stam1na - thrash metal, graveworm - black metal, kotiteollisuus - finnish, demons & wizards - Power metal, moonsorrow - folk metal, crematory - Gothic Metal, turmion

kätilöt - industrial metal, eternal tears of sorrow - Melodic Death Metal, leevi and the leavings - finnish, mokoma - thrash metal, insomnium - Melodic Death Metal, catamenia - melodic black metal, battlelore - folk metal, elvenking - folk metal, alestorm - pirate metal, eppu normaali - finnish, eisregen - black metal, swallow the sun - doom metal, falkenbach - viking metal, klamydia - punk, freedom call - Power metal, viikate - finnish, empyrium - doom metal, falconer - Power metal, corvus corax - medieval, diablo - Melodic Death Metal, thyrving - viking metal, eläkeläiset - humppa, the berzerker - grindcore, excrementory grindfuckers - grindcore, mors principium est - Melodic Death Metal, orphaned land - Progressive metal, before the dawn - Melodic Death Metal, poisonblack - Gothic Metal, faun - folk, disarmonia mundi - Melodic Death Metal, mánegarm - viking metal, vintersorg - folk metal, ajattara - black metal, arkona - folk metal, mercenary - Melodic Death Metal, 3 inches of blood - heavy metal, saltatio mortis - folk metal.

Musical World 11

Representative artist: autechre - idm

List: boards of canada - ambient, aphex twin - electronic, amon tobín - electronic, burial - dubstep, kraftwerk - electronic, brian eno - ambient, dead can dance - ambient, underworld - electronic, autechre - idm, squarepusher - idm, four tet - electronic, cocteau twins - dream pop, venetian snares - breakcore, prefuse 73 - electronic, shpongle - psychedelic, orbital - electronic, apparat - idm, the future sound of london - electronic, flying lotus - electronic, einstürzende neubauten - industrial, max richter - ambient, plaid - idm, tortoise - post-rock, coil - industrial, biosphere - ambient, the flashbulb - idm, lali puna - electronic, telefon tel aviv - idm, current 93 - neofolk, ellen allien - electronic, the orb - ambient, ellen allien & apparat -

electronic, stars of the lid - ambient, ulrich schnauss - ambient, john zorn - jazz, skream - dubstep, afx - idm, daedelus - electronic, the residents - experimental, ricardo villalobos - minimal, bengal - dubstep, helios - ambient, tim hecker - ambient, steve reich - minimalism, mum - electronic, nathan fake - minimal, fennesz - ambient, richie hawtin - minimal, clark - idm, talk talk - new wave, kettel - idm, arovane - idm, tycho - ambient, murcof - ambient, david sylvian - ambient, extrawelt - minimal, anthony rother - electro, ozric tentacles - Progressive rock, art of noise - electronic, bill hicks - comedy, mouse on mars - electronic, minilogue - minimal, scsi-9 - minimal, alex smoke - minimal, colleen - ambient, plastikman - minimal, enduser - breakcore, susumu yokota - ambient, cranes - shoegaze, boxcutter - dubstep, public image ltd. - post-punk, the field - minimal, throbbing gristle - industrial, isan - idm, wagon christ - electronic, gong - Progressive rock, deadbeat - dub, fanfare ciocărlia - Gypsy, monolake - minimal, atari teenage riot - digital hardcore, the klf - electronic, this mortal coil - ethereal, soft machine - Progressive rock, piano magic - post-rock, proem - idm, jóhann jóhannsson - ambient, luke vibert - idm, mujuice - idm.

Musical World 12

Representative artist: the game - rap

List: kanye west - Hip-Hop, eminem - rap, jay-z - Hip-Hop, nas - Hip-Hop, lil wayne - rap, 2pac - rap, 50 cent - rap, snoop dogg - Hip-Hop, t.i. - rap, lupe fiasco - Hip-Hop, notorious b.i.g. - Hip-Hop, dr. dre - rap, the game - rap, n*e*r*d - Hip-Hop, ludacris - rap, busta rhymes - Hip-Hop, ice cube - rap, flo rida - Hip-Hop, dmx - rap, dizzee rascal - Grime, chamillionaire - rap, nelly - Hip-Hop, r. kelly - rnb, three 6 mafia - rap, kid cudi - Hip-Hop, bone thugs-n-harmony - Hip-Hop, t-pain - rnb, prodigy - electronic, rick ross - rap, pitbull - Hip-Hop, eldo - Hip-Hop, fabolous - Hip-Hop, n.w.a -

Hip-Hop, the cool kids - Hip-Hop, clipse - Hip-Hop, citizen cope - indie, d12 - rap, tech n9ne - rap, soulja boy - Hip-Hop, kool savas - Deutschrapp, hilltop hoods - Hip-Hop, method man & redman - Hip-Hop, g-unit - rap, asa - Hip-Hop, flipsyde - Hip-Hop, ll cool j - Hip-Hop, insane clown posse - Horrorcore, camron - hip hop, ugk - Dirty South, centr - russian rap, lil jon & the east side boyz - Crunk, donguralesko - Hip-Hop, pharrell - Hip-Hop, lloyd banks - rap, trey songz - rnb, young jeezy - rap, dj khaled - Hip-Hop, fat joe - rap.

Musical World 13

Representative artist: epica - symphonic metal

List: nightwish - symphonic metal, apocalyptica - symphonic metal, sonata arctica - Power metal, anathema - doom metal, lacuna coil - Gothic Metal, kamelot - Power metal, therion - symphonic metal, epica - symphonic metal, paradise lost - Gothic Metal, ayreon - Progressive metal, moonspell - Gothic Metal, within temptation - Gothic Metal, blackmore's night - folk, avantasia - Power metal, the gathering - Gothic Metal, tiamat - Gothic Metal, rhapsody of fire - Power metal, pain of salvation - Progressive metal, symphony x - Progressive metal, theatre of tragedy - Gothic Metal, emilie autumn - Gothic, after forever - Gothic Metal, the 69 eyes - Gothic Rock, lordi - hard rock, lacrimosa - Gothic, tristania - Gothic Metal, haggard - symphonic metal, sirenia - Gothic Metal, angra - Power metal, tarja turunen - symphonic metal, trans-siberian orchestra - symphonic metal, leaves' eyes - Gothic Metal, evergrey - Progressive metal, rhapsody - Power metal, draconian - doom metal, vanessa-mae - instrumental, ost - Soundtrack, liquid tension experiment - Progressive metal, kittie - metal, e nomine - electronic, charon - Gothic Metal, dark moor - Power metal, entwine - Gothic Metal, lake of tears - doom metal, otep - Nu Metal, diablo swing

orchestra - Avant-garde Metal, delain - symphonic metal, lacrimas profundere - Gothic Metal, firewind - Power metal, lunatica - symphonic metal, edenbridge - symphonic metal, masterplan - Power metal, visions of atlantis - symphonic metal, the black mages - video game music, stream of passion - Progressive metal, inkubus sukkubus - Gothic Rock.

Musical World 14

Representative artist: simian mobile disco - electronic

List: the knife - electronic, justice - electronic, hot chip - electronic, ladytron - electronic, crystal castles - electronic, cansei de ser sexy - electronic, klaxons - new rave, cut copy - electronic, digitalism - electronic, lcd soundsystem - electronic, basement jaxx - electronic, róisín murphy - electronic, santogold - electronic, booka shade - minimal, trentemøller - minimal, simian mobile disco - electronic, the whitest boy alive - indie, death from above 1979 - indie, the presets - electronic, fischerspooner - electronic, the faint - indie, le tigre - indie, boys noize - electro, peaches - Electroclash, calvin harris - electronic, the gossip - indie, shiny toy guns - electronic, chromeo - electronic, mylo - electronic, ladyhawke - indie, high contrast - Drum and bass, mstrkrft - electronic, junior boys - electronic, mr. oizo - electronic, modeselektor - electronic, late of the pier - new rave, the rapture - indie, tiga - electronic, armand van helden - House, soulwax - electronic, justice - electronic, vitalic - electronic, new young pony club - electronic, gui boratto - minimal, empire of the sun - electronic, sébastien tellier - electronic, hercules and love affair - electronic, !!! - electronic, paul kalkbrenner - minimal, felix da housecat - electronic, friendly fires - electronic, m.i.a. - Hip-Hop, 2 many djs - electro, miss kittin - Electroclash, stereo total - electronic, la roux - electropop, vive la fête - electronic, midnight juggernauts - electronic, the horrors - Garage Rock, the teenagers -

indie, metronomy - electronic, uffie - electronic, datarock - electronic, lady sovereign - Grime, sam sparrow - , erlend Øye - electronic, cassius - House, the black ghosts - electronic, audio bullys - electronic, robots in disguise - Electroclash, just jack - electronic, fujiya & miyagi - electronic, dominik eulberg - minimal, annie - pop, matthew dear - electronic, noisia - Drum and bass, the organ - indie, birdy nam nam - turntablism, freestylers - breakbeat, little boots - electronic, sebastian - electro, martin solveig - House, goose - electronic, bonde do rolê - electronic, soko - french, de jeugd van tegenwoordig - Hip-Hop, van she - electronic, sascha funke - minimal, diplo - electronic, dragonette - electropop, laurent garnier - techno, yelle - french, dj mehdi - electro, lo-fi-fnk - electronic, pnau - electronic, spank rock - Hip-Hop, plump djs - breakbeat, the bloody beetroots - electro, happy mondays - madchester, glass candy - electronic.

Musical World 15

Representative artist: gang starr - Hip-Hop

List: the roots - Hip-Hop, outkast - Hip-Hop, cypress hill - Hip-Hop, a tribe called quest - Hip-Hop, common - Hip-Hop, rjd2 - Hip-Hop, atmosphere - Hip-Hop, jurassic 5 - Hip-Hop, mos def - Hip-Hop, de la soul - Hip-Hop, gang starr - Hip-Hop, jedi mind tricks - Hip-Hop, aesop rock - Hip-Hop, j dilla - Hip-Hop, mf doom - Hip-Hop, madlib - Hip-Hop, talib kweli - Hip-Hop, ghostface - Hip-Hop, madvillain - Hip-Hop, cunninlynguists - Hip-Hop, wu-tang clan - Hip-Hop, public enemy - Hip-Hop, immortal technique - Hip-Hop, mobb deep - Hip-Hop, roots manuva - Hip-Hop, fugees - Hip-Hop, sage francis - Hip-Hop, nujabes - Hip-Hop, quasimoto - Hip-Hop, method man - Hip-Hop, dilated peoples - Hip-Hop, gza/genius - Hip-Hop, looptroop - Hip-Hop, blackalicious - Hip-Hop, the pharcyde - Hip-Hop, q-tip - Hip-Hop, guru - Hip-Hop, rza - Hip-Hop, dangerdoom - Hip-Hop,

redman - Hip-Hop, big l - Hip-Hop, masta ace - Hip-Hop, little brother - Hip-Hop, raekwon - Hip-Hop, dead prez - Hip-Hop, run-d.m.c. - Hip-Hop, krs-one - Hip-Hop, murs - Hip-Hop, deltron 3030 - Hip-Hop, necro - Hip-Hop, people under the stairs - Hip-Hop, slum village - Hip-Hop, del tha funkee homosapien - Hip-Hop, buck 65 - Hip-Hop, krec - russian rap, brother ali - Hip-Hop, jaylib - Hip-Hop, el-p - Hip-Hop, k-os - Hip-Hop, black star - Hip-Hop, pete rock - Hip-Hop, ol dirty bastard - rap, jay dee - Hip-Hop, army of the pharaohs - Hip-Hop, az - Hip-Hop, mc solaar - french, pete philly & perquisite - Hip-Hop, ill bill - Hip-Hop, hieroglyphics - Hip-Hop, handsome boy modeling school - Hip-Hop, pharoahe monch - Hip-Hop, onyx - Hip-Hop, jeru the damaja - Hip-Hop, cut chemist - turntablism, iam - Hip-Hop.

Musical World 16

Representative artist: toska - downtempo

List: thievery corporation - chillout, bonobo - downtempo, morcheeba - trip-hop, dj shadow - trip-hop, faithless - electronic, groove armada - electronic, the cinematic orchestra - jazz, lamb - trip-hop, moloko - electronic, unkle - trip-hop, sia - female vocalists, gotan project - tango, tricky - trip-hop, hooverphonic - trip-hop, nightmares on wax - trip-hop, de-phazz - lounge, air - electronic, télépopmusik - electronic, toska - downtempo, zero 7 - chillout, dj krush - trip-hop, koop - jazz, kosheen - electronic, the herbaliser - Hip-Hop, st. germain - chillout, sneaker pimps - trip-hop, asian dub foundation - dub, everything but the girl - electronic, jazzanova - chillout, pink martini - jazz, kruder & dorfmeister - electronic, mr. scruff - electronic, parov stelar - nu-jazz, quantic - downtempo, flunk - trip-hop, blockhead - trip-hop, bent - chillout, nitin sawhney - chillout, dzihan & kamien - downtempo, skalpel - jazz, nouvelle vague - Bossa Nova,

leftfield - electronic, fila brazillia - chillout, death in vegas - electronic, boozoo bajou - downtempo, 4hero - electronic, saint etienne - electronic, coldcut - ninja tune, lemon jelly - electronic, kid loco - trip-hop, jaga jazzist - jazz, funki porcini - ninja tune, dj food - ninja tune, lemongrass - downtempo, dj vadim - ninja tune, bajofondo tango club - tango, dj cam - trip-hop, apollo 440 - electronic, mo horizons - downtempo, nicola conte - jazz, waldeck - trip-hop, brazilian girls - electronic, husky rescue - downtempo, david holmes - electronic, fun lovin criminals - funk, afterlife - chillout, gus gus - electronic, kid koala - turntablism, fink - indie, sofa surfers - downtempo, red snapper - trip-hop, fluke - electronic, martina topley-bird - trip-hop, lovage - trip-hop, smoke city - trip-hop, alice russell - soul, propellerheads - electronic, ltj bukem - Drum and bass, swayzak - electronic, beth gibbons & rustin man - trip-hop, planet funk - electronic, 9 lazy 9 - ninja tune, mandalay - trip-hop, the quantic soul orchestra - funk, aim - trip-hop, herbert - electronic, bitter:sweet - trip-hop, jazzamor - lounge, stéphane pompougnac - lounge, ott - dub, minus 8 - downtempo, lhasa - female vocalists, micatone - acid jazz, paris combo - french, stereo mcs - dance, miguel migs - deep house, bran van 3000 - electronic, nils petter molvær - jazz, kraak & smaak - electronic, le peuple de lherbe - trip hop, wax tailor - trip-hop.

Musical World 17

Representative artist: isis - Sludge

List: deftones - metal, porcupine tree - Progressive rock, faith no more - rock, katatonia - doom metal, god is an astronaut - post-rock, mastodon - Progressive metal, 65daysofstatic - post-rock, king crimson - Progressive rock, at the drive-in - post-hardcore, dredg - Progressive rock, converge - hardcore, meshuggah - Progressive metal, kyuss - Stoner Rock, primus -

alternative, the dillinger escape plan - mathcore, isis - Sludge, ...and you will know us by the trail of dead - indie, melvins - Grunge, riverside - Progressive rock, godspeed you! black emperor - post-rock, gojira - death metal, neurosis - Sludge, blackfield - Progressive rock, strapping young lad - industrial metal, mr. bungle - experimental, clutch - Stoner Rock, fantômas - experimental, mono - post-rock, down - stoner metal, a silver mt. zion - post-rock, cult of luna - Sludge, devin townsend - Progressive metal, boris - drone, 16 horsepower - Alt-country, cynic - Progressive metal, this will destroy you - post-rock, pig destroyer - grindcore, saul williams - Hip-Hop, antimatter - dark ambient, swans - industrial, oceansize - Progressive rock, fu manchu - Stoner Rock, pelican - post-rock, puscifer - industrial, bohren & der club of gore - jazz, jesu - post-rock, red sparowes - post-rock, mark lanegan - singer-songwriter, desert sessions - Stoner Rock, sparta - post-hardcore, melt-banana - noise rock, sunny day real estate - emo, shellac - math rock, the locust - grindcore, envy - screamo, earth - drone, team sleep - alternative, murder by death - indie, orchid - screamo, russian circles - post-rock, helmet - alternative metal, caspian - post-rock, don caballero - math rock, tomahawk - experimental, the black heart procession - indie, van der graaf generator - Progressive rock, genghis tron - cybergrind, mclusky - noise rock, the sound of animals fighting - experimental, grails - post-rock, the jesus lizard - noise rock, modern life is war - hardcore, screaming trees - Grunge, black mountain - Psychedelic Rock, tenhi - neofolk, peeping tom - experimental, novembre - doom metal, aereogramme - post-rock, pure reason revolution - Progressive rock, torche - Sludge, baroness - Sludge, naked city - Avant-Garde, green carnation - Progressive metal.

Musical World 18

Representative artist: all time low - pop punk

List: fall out boy - pop punk, paramore - rock, panic at the disco - alternative, the all-american rejects - rock, brand new - emo, the used - emo, taking back sunday - emo, dashboard confessional - emo, simple plan - rock, new found glory - pop punk, plain white ts - indie rock, anberlin - rock, gym class heroes - Hip-Hop, funeral for a friend - emo, saosin - post-hardcore, motion city soundtrack - pop punk, hellogoodbye - indie, the academy is... - emo, reliant k - rock, angels & airwaves - alternative, silverstein - screamo, jack's mannequin - piano rock, the red jumpsuit apparatus - emo, boys like girls - pop punk, something corporate - emo, say anything - indie, all time low - pop punk, saves the day - emo, metro station - electronic, senses fail - emo, third eye blind - rock, jonas brothers - pop, story of the year - emo, the get up kids - emo, cobra starship - pop punk, cute is what we aim for - pop punk, +44 - punk rock, escape the fate - post-hardcore, chiodos - post-hardcore, bowling for soup - punk rock, mae - emo, the ataris - punk, from first to last - screamo, sugarcult - rock, the hush sound - indie, mcfly - pop rock, copeland - indie, emery - screamo, yellowcard - punk rock, 3oh!3 - electronic, the format - indie, cartel - pop punk, augustana - alternative, the starting line - pop punk, finch - emo, secondhand serenade - acoustic, armor for sleep - emo, hawthorne heights - emo, mayday parade - pop punk, the rocket summer - indie, scary kids scaring kids - post-hardcore, the spill canvas - emo, jim sturgess - Soundtrack, set your goals - pop punk, jonathan larsen - Broadway, bayside - emo, dane cook - comedy, hollywood undead - rapcore, the early november - emo, howie day - singer-songwriter, forever the sickest kids - pop punk, phantom planet - rock, the string quartet - instrumental, aiden - post-hardcore, matt nathanson - singer-songwriter, box car racer - punk rock, kill hannah - alternative, head automatica - rock, four year strong - pop punk, acceptance - rock, the lonely island - comedy, owl city - electronic, straylight run - emo, manchester orchestra - indie, amber pacific - pop punk, demi lovato - pop, fightstar - post-hardcore, hannah montana - pop, busted - pop, the maine - pop punk, matchbook romance -

emo, hit the lights - pop punk, o.a.r. - rock, family force 5 - Crunk Rock, kate voegele - female vocalists, rufio - pop punk, brokencyde - crunkcore, zebrahead - punk rock, the almost - rock, american hi-fi - rock, madina lake - rock.

Musical World 19

Representative artist: above & beyond - trance

List: tiësto - trance, infected mushroom - psytrance, atb - trance, armin van buuren - trance, scooter - techno, paul van dyk - trance, the crystal method - electronic, ferry corsten - trance, paul oakenfold - trance, blank & jones - trance, above & beyond - trance, chicanne - trance, bt - trance, deadmau5 - House, junon reactor - electronic, gigi dagostino - House, london electricity - Drum and bass, oceanlab - trance, fragma - trance, 4 strings - trance, astral projection - psytrance, eric prydz - House, hybrid - electronic, deep dish - House, junkie xl - electronic, kaskade - House, robert miles - trance, modern talking - 80s, darude - trance, cosmic gate - trance, logistics - Drum and bass, milk inc. - dance, sylver - dance, 1200 micrograms - psytrance, markus schulz - trance, sash! - dance, kyau vs. albert - trance, sasha - trance, ministry of sound - electronic, ian van dahl - trance, rank 1 - trance, john dahlbäck - House, dj antoine - House, bomfunk mcs - Freestyler, astrix - psytrance, concord dawn - Drum and bass, blue man group - experimental, angerfist - hardcore, lasgo - dance, benassi bros. - House, solar stone - trance, iio - dance, pakito - dance, 2 unlimited - eurodance, gabriel & dresden - trance, pulsedriver - trance, marco v - trance, sunlounger - trance, armin van buuren feat. sharon den adel - trance, spor - Drum and bass, shapeshifters - House, east clubbers - dance, andain - trance, mauro picotto - trance.

Musical World 20

Representative artist: darkthrone - black metal

List: behemoth - death metal, cannibal corpse - death metal, ulver - black metal, satyricon - black metal, my dying bride - doom metal, napalm death - grindcore, burzum - black metal, carcass - death metal, nile - death metal, darkthrone - black metal, vader - death metal, type o negative - Gothic Metal, bathory - black metal, immortal - black metal, morbid angel - death metal, at the gates - Melodic Death Metal, emperor - black metal, samael - black metal, sodom - thrash metal, mayhem - black metal, marduk - black metal, deicide - death metal, municipal waste - thrash metal, gorgoroth - black metal, nasum - grindcore, death in june - neofolk, summoning - black metal, enslaved - black metal, six feet under - death metal, dissection - black metal, venom - black metal, entombed - death metal, obituary - death metal, bolt thrower - death metal, arcturus - black metal, rotting christ - black metal, electric wizard - doom metal, carpathian forest - black metal, belphegor - black metal, necrophagist - Technical Death Metal, anaal nathrakh - black metal, celtic frost - black metal, aborted - death metal, cryptopsy - death metal, decapitated - death metal, candlemass - doom metal, dying fetus - death metal, suffocation - death metal, shining - black metal, borknagar - black metal, dark funeral - black metal, impaled nazarene - black metal, dismember - death metal, lifelover - black metal, cavalera conspiracy - thrash metal, drudkh - black metal, anal cunt - grindcore, unleashed - death metal, 1349 - black metal, rotten sound - grindcore, primordial - black metal, septic flesh - death metal, atheist - death metal, d.r.i. - crossover, isis - Sludge, old man's child - black metal, discharge - d-beat, naglfar - black metal, deathspell omega - black metal, taake - black metal, nokturnal mortum - black metal.

Musical World 21

Representative artist: parkway drive - metalcore

List: killswitch engage - metalcore, as i lay dying - metalcore, lamb of god - metalcore, underoath - screamo, alexisonfire - post-hardcore, trivium - metalcore, thrice - post-hardcore, enter shikari - post-hardcore, all that remains - metalcore, hatebreed - hardcore, bring me the horizon - deathcore, a day to remember - post-hardcore, city and colour - acoustic, thursday - post-hardcore, devildriver - metalcore, heaven shall burn - metalcore, comeback kid - hardcore, parkway drive - metalcore, circa survive - indie, the fall of troy - post-hardcore, chimaira - metalcore, 36 crazyfists - metalcore, caliban - metalcore, the black dahlia murder - death metal, from autumn to ashes - metalcore, alesana - post-hardcore, all shall perish - deathcore, the devil wears prada - metalcore, between the buried and me - Progressive metal, the haunted - thrash metal, horse the band - Nintendocore, august burns red - metalcore, ignite - hardcore, protest the hero - metalcore, walls of jericho - hardcore, norma jean - hardcore, every time i die - hardcore, have heart - hardcore, unearth - metalcore, suicide silence - deathcore, haste the day - metalcore, darkest hour - metalcore, terror - hardcore, job for a cowboy - deathcore, madball - hardcore, andy mckee - acoustic, bleeding through - metalcore, blessthefall - post-hardcore, despised icon - deathcore, evergreen terrace - hardcore, poison the well - hardcore, i killed the prom queen - metalcore, as blood runs black - deathcore, glassjaw - post-hardcore, boyssetfire - hardcore, disco ensemble - rock, dance gavin dance - post-hardcore, fear before the march of flames - hardcore, neaera - metalcore, verse - hardcore, shadows fall - metalcore, ceremony - hardcore, emmure - metalcore, a skylit drive - post-hardcore, carnifex - deathcore, misery signals - metalcore, eyes set to kill - post-hardcore, deadlock - Melodic Death Metal, the number twelve looks like you - mathcore, champion - hardcore, it dies today - metalcore, bury your

dead - hardcore, a static lullaby - post-hardcore, bane - hardcore, stigmata - metalcore, blood for blood - hardcore, biohazard - hardcore, architects - metalcore, amatory - metalcore, in this moment - metalcore, still remains - metalcore, dry kill logic - metalcore, he is legend - hardcore, roadrunner united - metal, gallows - hardcore, murderdolls - horror punk, ektomorf - thrash metal, blacklisted - hardcore, sabrepulse - 8-bit, emarosa - post-hardcore.

Musical World 22

Representative artist: bill evans - jazz

List: miles davis - jazz, billie holiday - jazz, ella fitzgerald - jazz, louis armstrong - jazz, john coltrane - jazz, diana krall - jazz, jamie cullum - jazz, herbie hancock - jazz, madeleine peyroux - jazz, chet baker - jazz, nat king cole - jazz, django reinhardt - jazz, thelonious monk - jazz, duke ellington - jazz, charlie parker - jazz, bill evans - jazz, esbjörn svensson trio - jazz, glenn gould - Classical, charles mingus - jazz, dean martin - jazz, keith jarrett - jazz, astor piazzolla - tango, sarah vaughan - jazz, pat metheny - jazz, james taylor - folk, chick corea - jazz, stan getz - jazz, dave brubeck - jazz, glenn miller - jazz, julie london - jazz, dinah washington - jazz, gonzales - piano, count basie - jazz, squirrel nut zippers - swing, oscar peterson - jazz, the dave brubeck quartet - jazz, john scofield - jazz, peggy lee - jazz, george benson - jazz, ry cooder - blues, benny goodman - jazz, pizzicato five - japanese, paolo conte - jazz, sun ra - jazz, cassandra wilson - jazz, ella fitzgerald & louis armstrong - jazz, al di meola - jazz, jaco pastorius - jazz, marcus miller - jazz, stacey kent - jazz, barbra streisand - female vocalists, penguin cafe orchestra - instrumental, pat metheny group - jazz, kronos quartet - Classical, sonny rollins - jazz, tony bennett - jazz, mahavishnu orchestra - jazz, cannonball adderley - jazz, the bad plus - jazz,

tower of power - funk, room eleven - jazz, brad mehldau - jazz, chris botti - jazz, medeski, martin and wood - jazz, く る り - japanese, john martyn - folk, béla fleck and the flecktones - jazz, erik truffaz - jazz, bobby mcferrin - jazz, ibrahim ferrer - cuban, mr.children - j-pop, bill frisell - jazz, ali farka touré - african, anouar brahem - jazz, vince guaraldi trio - jazz, dizzy gillespie - jazz.

Musical World 23

Representative artist: marisa monte - mpb

List: Édith piaf - french, carla bruni - french, caetano veloso - mpb, chico buarque - mpb, los hermanos - brazilian, marisa monte - mpb, bebel gilberto - Bossa Nova, antônio carlos jobim - Bossa Nova, simply red - pop, os mutantes - psychedelic, astrud gilberto - Bossa Nova, seu jorge - brazilian, orishas - latin, legião urbana - rock, joão gilberto - Bossa Nova, adriana calcanhotto - mpb, jorge ben - mpb, elis regina - mpb, vanessa da mata - mpb, cesária Évora - female vocalists, donavon frankenreiter - acoustic, jorge drexler - cantautor, engenheiros do hawaii - rock, cássia eller - mpb, gilberto gil - brazilian, little joy - indie, raul seixas - rock, paco de lucía - Flamenco, madredeus - fado, adele - soul, maria rita - mpb, cartola - samba, skank - rock, lenine - mpb, djavan - mpb, sérgio mendes - Bossa Nova, charlie brown jr. - rock, céu - mpb, gal costa - Bossa Nova, capital inicial - rock, pato fu - rock, cazuza - mpb, matanza - countrycore, charles aznavour - french, os paralamas do sucesso - rock, tim maia - soul, o rappa - rock, charly garcía - Rock Argentino, ana carolina - mpb, raimundos - hardcore, tom zé - tropicalia, rita lee - mpb, maria bethânia - mpb, milton nascimento - mpb, pitty - rock, o teatro mágico - mpb.

Musical World 24

Representative artist: pidżama porno - punk rock

List: myslovitz - polish, o.s.t.r. - Hip-Hop, happysad - rock, hey - rock, kult - rock, pidżama porno - punk rock, coma - rock, 5'nizza - reggae, akurat - ska, sinéad o'connor - female vocalists, dzem - blues, vavamuffin - reggae, t.love - rock, maria peszek - polish, skunk anansie - rock, jamal - reggae, glen hansard - singer-songwriter, blue foundation - trip-hop, kazik - polish, kaliber 44 - psycho rap, paktofonika - Hip-Hop, lao che - alternative, balkan beat box - balkan, louise attaque - french, czesław Śpiewa - alternative, emir kusturica & the no smoking orchestra - balkan, farben lehre - punk rock, marek grechuta - polish, smolik - chillout, nosowska - polish, republika - rock, anja garbarek - trip-hop, pezet - Hip-Hop, stare dobre małżeństwo - poezja spiewana, indios bravos - reggae, grzegorz turnau - poetry, waglewski fisz emade - alternative, pati yang - trip-hop, habakuk - reggae, the analogs - Oi, hurt - rock, Łona - Hip-Hop, poe - female vocalists, hunter - heavy metal, strachy na lachy - rock, natural dread killaz - reggae, anna maria jopek - jazz, joe dassin - french, cool kids of death - polish.

Musical World 25

Representative artist: covenant - ebm

List: pet shop boys - 80s, a-ha - 80s, siouxsie and the banshees - post-punk, bauhaus - post-punk, vnv nation - ebm, the sisters of mercy - Gothic Rock, ministry - industrial, she wants revenge - post-punk, combichrist - industrial, apocrygma berzerk - ebm, iamx - electronic, kmfdm - industrial, tears for fears - 80s, skinny puppy - industrial, the birthday massacre - industrial, erasure - 80s, covenant - ebm, blutengel - darkwave, l'Âme immortelle - darkwave, front 242 - ebm, and one - synthpop, orchestral manoeuvres in the dark - new wave, simple minds - 80s, diary of dreams - darkwave, front

line assembly - industrial, laibach - industrial, assemblage 23 - ebm, vast - alternative, the human league - new wave, deine lakaien - darkwave, suicide commando - ebm, alphaville - 80s, hocico - ebm, rasputina - cello rock, dave gahan - electronic, the crüxshadows - darkwave, unheilig - Gothic, killing joke - post-punk, clan of xymox - darkwave, gary numan - new wave, grendel - ebm, sopor aeternus & the ensemble of shadows - darkwave, de/vision - synthpop, london after midnight - Gothic Rock, alice in videoland - electronic, christian death - deathrock, qntal - medieval, freezepop - synthpop, funker vogt - ebm, eisbrecher - industrial, das ich - industrial, orgy - industrial, :wumpscut: - industrial, ultravox - new wave, project pitchfork - industrial, psyclon nine - industrial, the chameleons - post-punk, velvet acid christ - industrial, client - electronic, soft cell - new wave, dope stars inc. - industrial, fields of the nephilim - Gothic Rock, feindflug - industrial, new model army - post-punk, agonoize - ebm, icon of coil - ebm, indochine - french, god module - ebm, the legendary pink dots - experimental, mesh - synthpop, camouflage - synthpop, colony 5 - futurepop, nitzer ebb - ebm, ayria - ebm.

Musical World 26

Representative artist: jill scott - soul

List: stevie wonder - soul, prince - funk, erykah badu - soul, marvin gaye - soul, john legend - soul, aretha franklin - soul, sade - soul, joss stone - soul, al green - soul, corinne bailey rae - soul, lauryn hill - soul, mary j. blige - rnb, aaliyah - rnb, curtis mayfield - soul, jill scott - soul, barry white - soul, india. arie - soul, sly & the family stone - funk, jamie lidell - soul, ayo - soul, the jackson 5 - soul, amos lee - singer-songwriter, kelis - rnb, brandy - rnb, etta james - blues, the temptations - soul, bill withers - soul, macy gray - soul, estelle - rnb, incognito - acid jazz, robin thicke - soul, funkedelic -

funk, earth, wind & fire - funk, dangelo - , the brand new heavies - acid jazz, kool & the gang - funk, fat freddy's drop - dub, angie stone - soul, toni braxton - rnb, amerie - rnb, nneka - soul, keyshia cole - rnb, james brown - funk, raphael saadiq - soul, grace jones - Disco, the isley brothers - soul, tlc - rnb, boyz ii men - rnb, natalie merchant - female vocalists, isaac hayes - soul, maxwell - soul, musiq - soul, anthony hamilton - soul, npr - podcast, diana ross - soul, fela kuti - afrobeat, the meters - funk, sharon jones and the dap-kings - soul, ben harper & the innocent criminals - rock, us3 - acid jazz, parliament - funk, dwele - Neo-Soul, gil scott-heron - soul, blue six - deep house, the four tops - soul, roy ayers - funk.

Musical World 27

Representative artist: håkan hellström - swedish

List: kent - swedish, lykke li - swedish, the sounds - indie, anna ternheim - singer-songwriter, shout out louds - indie, the radio dept. - shoegaze, joshua radin - acoustic, the hellacopters - rock, håkan hellström - swedish, eva cassidy - female vocalists, glasvegas - indie, lars winnerbäck - swedish, slagsmålsklubben - electronic, ane brun - singer-songwriter, hello saferide - swedish, the magic numbers - indie, dolly parton - country, françoise hardy - french, josh rouse - singer-songwriter, kaizers orchestra - norwegian, johnessi - swedish, the ark - swedish, i'm from barcelona - indie pop, tiger lou - swedish, caesars - indie, timbuktu - swedish, billie the vision & the dancers - swedish, emmylou harris - country, lisa ekdahl - jazz, sophie zelmani - singer-songwriter, veronica maggio - swedish, cornelis vreeswijk - swedish, france gall - french, stina nordenstam - female vocalists, el perro del mar - swedish, bo kaspers orkester - swedish, detektivbyrån - instrumental, lisa miskovsky - pop, the tough alliance - swedish, sahara hotnights - rock, the perishers - indie, takida - rock, moneybrother -

swedish, familjen - swedish, bob hund - swedish, laleh - swedish, thomas dybdahl - singer-songwriter, dungen - psychedelic, club 8 - swedish, kloorup - electronic, lene marlin - pop, ronan keating - pop, mustasch - Stoner Rock, april march - female vocalists, miss li - swedish, acid house kings - indie pop, ebba grön - punk, broder daniel - swedish, teitur - singer-songwriter, the soundtrack of our lives - swedish.

Musical World 28

Representative artist: the pigeon detectives - indie

List: kaiser chiefs - indie, editors - indie, kasabian - indie, kate nash - female vocalists, the libertines - indie, the fratellis - indie, maximo park - indie, razorlight - indie, babyshambles - indie, the last shadow puppets - indie, foals - indie, the wombats - indie, jet - rock, the subways - indie rock, paolo nutini - singer-songwriter, the bravery - indie, ok go - indie, hot hot heat - indie, feeder - rock, tokyo police club - indie rock, biffy clyro - rock, mark ronson - funk, hard-fi - indie, dirty pretty things - indie, athlete - indie, los campesinos! - indie pop, the vines - rock, the coral - indie, the pigeon detectives - indie, the rakes - indie, the cribs - indie, the pipettes - indie pop, laura marling - folk, the futureheads - indie, art brut - indie, black kids - indie, the maccabees - indie, alphabeat - pop, the zutons - indie, hadouken! - new rave, ash - rock, rooney - indie, mystery jets - indie, jamie t - indie, ian brown - indie, newton faulkner - acoustic, the feeling - indie, the view - indie, idlewild - indie, be your own pet - indie, the charlatans - britpop, the lemonheads - indie, blood red shoes - indie rock, noah and the whale - folk, guillemots - indie, we are scientists - indie, the long blondes - indie, good shoes - indie, the beautiful south - pop, air traffic - indie, the hoosiers - indie, florence + the machine - indie, embrace - britpop, richard ashcroft - britpop, scouting for girls - indie, turin brakes - indie, the rifles - indie, the

automatic - indie, the music - rock, get cape. wear cape. fly - indie, ocean colour scene - britpop, the thrills - indie, frank turner - folk, peter doherty - indie, sugarplum fairy - swedish.

Musical World 29

Representative artist: la oreja de van gogh - spanish

List: shakira - pop, various artists - heavy metal, the corrs - pop, juanes - latin, café tacuba - rock, soda stereo - Rock Argentino, andrés calamaro - Rock Argentino, héroes del silencio - rock, la oreja de van gogh - spanish, maná - latin, julieta venegas - latin, los fabulosos cadillacs - ska, daddy yankee - Reggaeton, babasónicos - Rock Argentino, joaquín sabina - cantautor, laura pausini - italian, m̄ago de oz - folk metal, gustavo cerati - Rock Argentino, extremoduro - rock, jarabe de palo - spanish, amaral - spanish, eros ramazzotti - italian, los planetas - indie, fito & fitipaldis - Spanish Rock, molotov - rapcore, alizée - french, chambao - Flamenco, texas - pop, ojos de brujo - Flamenco, el canto del loco - spanish, enrique bunbury - rock, tiziano ferro - italian, alejandro sanz - latin, belanova - electropop, fito p̄az - Rock Argentino, ricardo arjona - latin, russian red - folk, pereza - spanish, bebe - spanish, macaco - Fusion, ricky martin - latin, giovanotti - italian, vetusta morla - indie rock, silvio rod̄ríguez - trova, estopa - spanish, don omar - Reggaeton, la casa azul - indie pop, marea - rock, calle 13 - Reggaeton, marc anthony - salsa, los piratas - indie, mecano - pop, dover - rock, love of lesbian - indie, thirteen senses - indie, james morrison - singer-songwriter, lori meyers - indie, gloria estefan - latin, freddie mercury - classic rock, bersuit vergarabat - Rock Argentino, aterciopelados - Colombia, bonnie tyler - 80s, il divo - Classical, caifanes - Rock en Espanol, aventura - bachata, nena daconte - spanish, luis miguel - boleros, enanitos verdes - Rock en Espanol, rbd - pop, bond - instrumental, sin bandera -

Romantica, los rodríguez - Rock Argentino, lisa loeb - female vocalists, alejandro fernández - latin, deluxe - indie.

Musical World 30

Representative artist: amethystium - ambient

List: enya - new age, enigma - new age, mike oldfield - Progressive rock, vangelis - electronic, loreena mckennitt - celtic, delerium - ambient, jean-michel jarre - electronic, era - new age, sarah brightman - female vocalists, the alan parsons project - Progressive rock, deep forest - new age, schiller - electronic, secret garden - new age, clannad - celtic, café del mar - chillout, andrea bocelli - Classical, flëur - cardio-wave, amethystium - ambient, mylène farmer - french, kitaro - new age, boney m. - Disco, gregorian - new age, conjure one - ambient, dolphin - alternative, carbon based lifeforms - ambient, enigma - new age, tangerine dream - electronic, Сплин - russian rock, vibrasphere - psytrance, karunesh - new age, lisa gerrard - ethereal, земфира - russian rock, hallucinogen - psytrance, cirque du soleil - ambient, nautilus pompilius - russian rock, solar fields - ambient, mediæval bæbes - medieval, arcana - dark ambient, celtic woman - celtic, david arkenstone - new age, kenny g - jazz, adiemus - new age, hammock - post-rock, sandra - 80s, entheogenic - ambient, bliss - chillout, Кино - russian rock, the chieftains - celtic, bang gang - trip-hop, keiko matsui - jazz, acoustic alchemy - Smooth Jazz, ravi shankar - Indian, e.s. posthumus - instrumental, kim wilde - 80s, bluetech - ambient, banco de gaia - ambient.

Musical World 31

Representative artist: sizzla - reggae

List: matisyahu - reggae, damian marley - reggae, gentleman - reggae, sean paul - dancehall, patrice - reggae, toots and the maytals - reggae, ub40 - reggae, easy star all-stars - reggae, aphrodite - Drum and bass, sizzla - reggae, peter tosh - reggae, augustus pablo - dub, burning spear - reggae, black sun empire - Drum and bass, lee "scratch" perry - dub, mattafix - Hip-Hop, chase & status - Drum and bass, shaggy - reggae, tryo - reggae, promoe - Hip-Hop, groundation - reggae, king tubby - dub, black uhuru - reggae, skindred - reggae metal, anthony b - reggae, capleton - reggae, buju banton - reggae, desmond dekker - ska, barrington levy - reggae, gregory isaacs - reggae, jimmy cliff - reggae, alborosie - reggae, beenie man - dancehall, collie buddz - reggae, alpha blondy - reggae, horace andy - reggae, eek-a-mouse - reggae, israel vibration - reggae, dennis brown - reggae, steel pulse - reggae, scientist - dub, max romeo - reggae, dj fresh - Drum and bass.

Musical World 32

Representative artist: franz schubert - Classical

List: johann sebastian bach - Classical, ludwig van beethoven - Classical, frédéric chopin - Classical, antonio vivaldi - Classical, wolfgang amadeus mozart - Classical, erik satie - Classical, pyotr ilyich tchaikovsky - Classical, claude debussy - Classical, georg friedrich händel - Classical, jacques brel - french, johannes brahms - Classical, franz schubert - Classical, arvo pärt - Classical, sergei rachmaninoff - Classical, edvard grieg - Classical, antonín dvořák - Classical, franz joseph haydn - Classical, adriano celentano - italian, giuseppe verdi - Classical, the tiger lillies - dark cabaret, felix mendelssohn - Classical, franz liszt - Classical, Ólafur arnalds - ambient, georges brassens - chanson francaise, carl orff - Classical, igor

stravinsky - Classical, caparezza - italian, richard wagner - Classical, patsy cline - country, luciano pavarotti - opera.

Musical World 33

Representative artist: ガゼット - J-rock

List: dir en grey - J-rock, maximum the hormone - J-rock, l'arc~en~ciel - J-rock, asian kung-fu generation - J-rock, the pillows - J-rock, boa - j-pop, apulanta - finnish, gackt - J-rock, tokio hotel - rock, ガゼット - J-rock, ムック - J-rock, capsule - electronic, 宇多田ヒカル - j-pop, perfume - electronic, déspairsray - J-rock, orange range - J-rock, 浜崎あゆみ - j-pop, uverworld - J-rock, shiro sagisu - Soundtrack, girugamesh - J-rock, high and mighty color - J-rock, jack off jill - female vocalists, polysics - japanese, m-flo - japanese, malice mizer - visual kei, 동방신기 - k-pop, ellegarden - J-rock, nightmare - J-rock, scandinavian music group - finnish, kotoko - j-pop, do as infinity - j-pop, miyavi - J-rock, epik high - Korean, big bang - k-pop.

Musical World 34

Representative artist: james newton howard - Soundtrack

List: hans zimmer - Soundtrack, howard shore - Soundtrack, john williams - Soundtrack, ennio morricone - Soundtrack, clint mansell - Soundtrack, danny elfman - Soundtrack, philip glass - minimalism, thomas newman - Soundtrack, klaus badelt - Soundtrack, andrew lloyd webber - musical, ludovico einaudi - piano, michael nyman - Soundtrack, harry gregson-williams - Soundtrack, 植松伸夫 - Soundtrack, james horner - Soundtrack, angelo badalamenti - Soundtrack, 菅野よう子 - Soundtrack, carter burwell - Soundtrack, rob dougan - electronic, craig armstrong - Soundtrack, james newton howard - Soundtrack, a.r. rahman - Soundtrack, michael andrews -

Soundtrack, jonathan coulton - comedy, gary jules - singer-songwriter, gustavo santaolalla - Soundtrack, tyler bates - Soundtrack, hans zimmer & james newton howard - Soundtrack, michael giacchino - Soundtrack, bear mccreary - Soundtrack, 久石譲 - Soundtrack, john murphy - Soundtrack, alan silvestri - Soundtrack, vienna teng - female vocalists, john powell - Soundtrack, george winston - piano, alison krauss - bluegrass, jeremy soule - Soundtrack, the dust brothers - electronic, henry mancini - Soundtrack, jerry goldsmith - Soundtrack, alison krauss & union station - bluegrass, randy newman - singer-songwriter, london symphony orchestra - Classical, john barry - Soundtrack, dario Marianelli - Soundtrack, steve jablonsky - Soundtrack, frank klepacki - industrial, jaromír nohavica - folk, david arnold - Soundtrack, bruno coulais - Soundtrack, nick cave & warren ellis - Soundtrack, eric serra - Soundtrack, powderfinger - rock

Musical World 35

Representative artist: teoman - turkish rock

List: goran bregović - balkan, starsailor - indie, gipsy kings - latin, noir désir - french, reamonn - rock, sezen aksu - turkish, jay-jay johanson - trip-hop, duman - turkish rock, teoman - turkish rock, shantel - balkan, mor ve Ötesi - turkish rock, Şebnem ferah - turkish rock, pinhani - turkish rock, tarkan - turkish, oi va voi - trip-hop, vega - turkish rock, dolores o'riordan - female vocalists, chris de burgh - pop, sertab erener - turkish, mariza - fado.

Musical World 36

Representative artist: chris tomlin - christian

List: switchfoot - rock, mute math - indie, rascal flats - country, jars of clay - christian, sixpence none the richer - pop, hillsong united - christian,

Appendix B: 36 Musical Worlds according to Boolean AP

hillsong - christian, garth brooks - country, brad paisley - country, chris tomlin - christian, colin hay - acoustic, bing crosby - jazz, david crowder band - christian, kutless - christian rock, third day - christian, delirious? - christian, brett dennen - folk, michael w. smith - christian, casting crowns - christian, alan jackson - country, newsboys - christian, tim mcgraw - country, jeremy camp - christian, kenny chesney - country, pillar - christian rock.