

Project Semantic Web:

Human vs machine clustering in music collections

Ronald Chu, 9919198
Lieuwe Rekker, F093256
Rudy Boonekamp, 3174131

Keywords: semantic web, clustering, music

1. Introduction

Content tagging is widely being used to allow users to provide meta-data about content such as images, blog posts and music. Utilizing these tags, content can be grouped into meaningful clusters. On the website of Last.fm, user tagging is used to allow a user to add personal associations to a certain track, album or artist. Last.fm is a music tracking website that uses the music listening history of a user to create a profile and link the user to relevant music or other users. As such, tags and the associated clusters on this website will likely represent musical genres. Clustering by user tags is what will be referred to as human clustering.

However, there is a downside to human clustering. Tags with the same semantic value often differ in syntax creating redundant clusters. The amount of tags may differ depending on the subject creating a false sense of significance for this subject.

Therefore it is desirable to determine if there is another way of clustering music apart from tags. More specifically, the question is whether it is possible to cluster music using musical features extracted from the music track. Furthermore, it is of interest to determine whether these groups can be linked to a certain genre. If so, this could be a welcome addition or control tool for the user tagging system.

In this paper, clustering using tags and musical features will be compared to answer the above questions.

1.1. *Related research*

In this chapter, a research question will be formulated based on existing research.

At the moment, a lot of research concentrates on the clustering of tags to generate ontologies. In the research of Levy and Sandler (2007) tags are investigated as semantic metadata for music. Despite the ad-hoc and informal language of tags, tags represent musical genres very well. There are two important applications of this meta-data:

- driving search applications for similar music or sounds
- training audio classification systems by using the meta-data as a source of groundtruth

There are however several problems with tag based ontologies.

- Tags are often ambiguous. Users might misspell tags, include punctuation, spaces or use stemming variants. (Specia and Motta, 2007)
- Resources often do not have enough tags for reliable clustering.
- Tags are often highly informal and not necessarily related to the audio content. An example is a tag like 'frickin awesome'. (Levy and Sandler, 2007).
- Users might be biased when tagging music. A user might tag a track according to their expectations of an artist rather than the audio content of the specific track. For example, a user might tag a song of the grunge band Nirvana as grunge, while the specific song is very down tempo and has more resemblance to classical music.

A possible solution to this is an ontology based on system clustering, in this case by musical features derived from actual audio tracks. The uses for tag clustering mentioned above are also relevant for tags derived from musical feature clusters. The advantage of using a machine generated cluster over a tag cluster is that the machine generated cluster does not suffer from the cold start problem. Furthermore it is certain that suggestions are correct, unlike suggestions that are derived from a social web that might be ambiguous themselves. Therefore it is of interest to see if a comparable ontology can be generated using machine clustering in contrast to tag clustering. This ontology would enjoy the benefits of machine clustering mentioned above.

1.2. Research question

Based on the given research, the following research question is formulated:

'How does machine clustering by content compare to the clustering of social web meta-data in the domain of music?'

In order to answer the research question, it is important that machine clustering performs sufficiently. In this research, machine clustering is performed by analyzing musical features. In the 2005 MIREX contest for audio classification musical features have been successfully analyzed and clustered to form musical genre's. Because the same musical features will be used in this research, the first hypothesis is as follows:

Hypothesis 1: 'By utilizing musical features, it is possible to cluster music by similarity to form a genre'

To test this hypothesis music will be clustered by similarity using audio features. The tracks in the resulting clusters will be judged on uniformity to a musical genre.

Hypothesis 2: 'Clustering by musical features results in a similar network as clustering by social web meta-data'

To test this hypothesis the clustering result will be compared to clustering using tags. The tags are sourced from Last.fm (social web) and interpreted using the Last.fm API and clustering algorithms (semantic web).

2. Experiment design

2.1. *Methodology*

A music database is prepared consisting of a diverse collection of music tracks representing the most prominent genre's in music.

From all tracks, tags are read and the tracks are clustered by tag similarity.

From all tracks, the musical features of RP, RH and SSD are determined. The tracks are clustered by these musical features. The clusters are labeled by determining the top-10 of most frequent tags in the cluster.

The two resulting cluster collections are compared by tags, structure and possibly other discrepancies or similarities that become obvious. Furthermore, due to the fact that clustering runs utilize a random initial point, each clustering run is done twice and compared. This is a control for the stability of the algorithm.

From this comparison, it is determined what the advantages are of using machine clustering over the use of the social web. Possible implementations of this functionality in the social- and semantic web are proposed.

2.2. *Dataset*

Obtaining a large broad dataset is a common problem in the field of MIR. Because most music is copyrighted such a set cannot be shared among scientists and no open dataset is available. Using our own music collection we were able to compose a dataset of 592 tracks (with roughly 400 different artists) distributed over 8 main music genres (58 blues, 76 classical, 100 country, 45 drum & bass, 40 electro, 31 hip hop, 57 house, 55 metal, 99 Dutch folk, 40 rock and 7 uncommon (breakcore, chiptunes)).

3. Tags

In this chapter, the use of tags from Last.fm is described: how these were retrieved and used for clustering. Last.fm is an example of a broad folksonomy: many users tag the same resource. Therefore, tags express the judgement of multiple (preferably many) users and this represents a meaningful source of semantic meta-data.

Our handling of tags is based on the approach described by Specia and Motta. (2007) This is an approach where tags are preprocessed to disambiguate them. This can be done by checking morphological similarity and excluding isolated tags. We have done some disambiguation manually by merging tags from tracks that are actually the same. (For reasons unknown, Last.fm contains duplicate records of some tracks)

The second step in this approach is to cluster the tags statistically. Our statistical approach for tag clustering is described in the next chapter.

The last step in Motta's approach is relation identification by looking up terms in online ontologies. While we do not define the relations between clusters, we do label them according to a look-up in a online folksonomy: Last.fm.

Monachesi (2010) suggested crawling social media applications for ontology enrichment. In this paper, this is applied by using tags sourced from the Last.fm API to name clusters based on musical features.

The extensive API [9] of Last.fm and the python-lastfm [10] project enabled us to fairly easy retrieve the tags the users of Last.fm added to the tracks in our dataset. The main problem we encountered was the identification of tracks on two strings: artist and track. Since these values are not unique and there are many variations in spelling and notation (e.g. "bach", "Bach", "J.S. Bach", "Bach, Johann Sebastian") it turned out to be far from trivial to obtain the correct track on Last.fm. Another result of this problem is that there exist multiple entries on Last.fm for one track, each with it's own (often lack of) tracks. We improved the tag retrieval process by using the API's search functions for both artist and track to obtain the best result. For about 100 tracks the correct Last.fm entry was selected manually using a helper script to search and select a track.

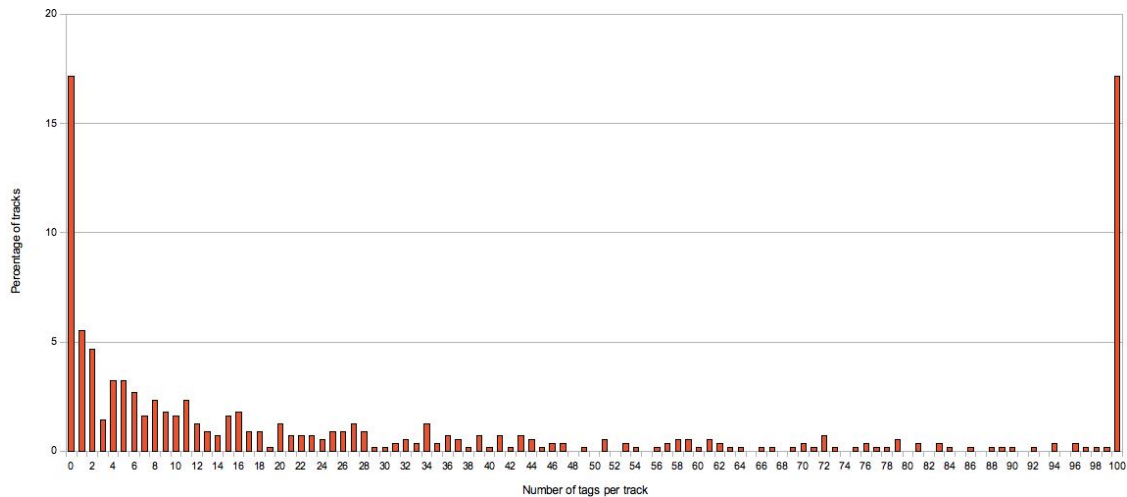


Figure 1

The resulting set of tags is somewhat disappointing. Out of the 592 tracks in the dataset 32 were not found or had no tags, the amount of tags returned for the remaining 560 tracks is depicted in [figure 1].

4. Musical features

In order to cluster music by the music signal itself, it is necessary to determine musical features that can both be used in a meaningful way to differentiate music and that can be extracted using an existing software tool. This has narrowed down the possible musical features for use.

Rhythm Patterns (RP)

Rhythm patterns describe fluctuations of the amplitude on a number of frequency bands that are critical to the human auditory system. By describing these fluctuations, this feature tries to capture that what humans perceive as rhythm. The result of this feature is a graph that shows for each critical band the intensity of the modulation frequency. In this way, the intensity of a certain rhythm in a certain frequency band is described. This is a

indication of the type of music; specific musical genre's show characteristic graphs with hotspots in specific band/modulation combinations. It is not our intention to specifically determine musical genre's as clusters but this does show that this feature is a descriptor of musical similarity.

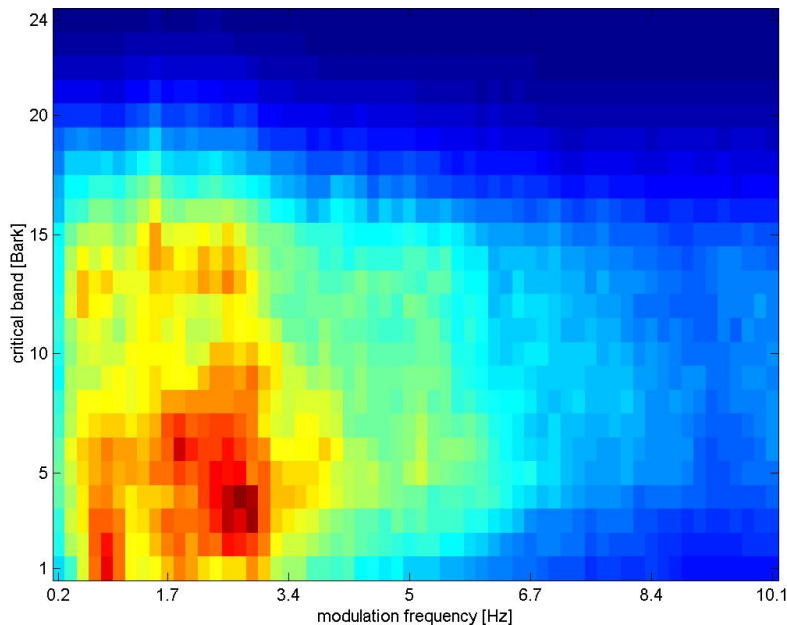


Figure 2

Statistical Spectrum Descriptor (SSD)

This feature consists of statistical measures that are applied to the sonogram calculated in the rhythm pattern. The sonogram reflects human loudness sensation in a power spectrum over frequency bands. Statistical measures that are applied to the critical bands are:

- mean loudness
- median loudness
- variance of loudness
- skewness of loudness (asymmetry of the probability distribution)
- kurtosis (amount of extremes in loudness causing variance)
- minimum and maximum value of loudness

These statistical measures are not likely to describe musical similarity by themselves as they are 1-dimensional and represent a very low-level analysis of musical features. However, the combination of these statistical descriptors can be characteristic for musical similarity.

Rhythm Histogram (RH)

The rhythm histogram is formed by summing up all critical bands for each modulation frequency. This results in a histogram portraying the rhythmic energy per modulation frequency. These frequencies are shown as 60 bins between 0 and 10hz. Then, the mean value of each bin for the music track is calculated. This feature is a descriptor for general rhythmic in an audio document.

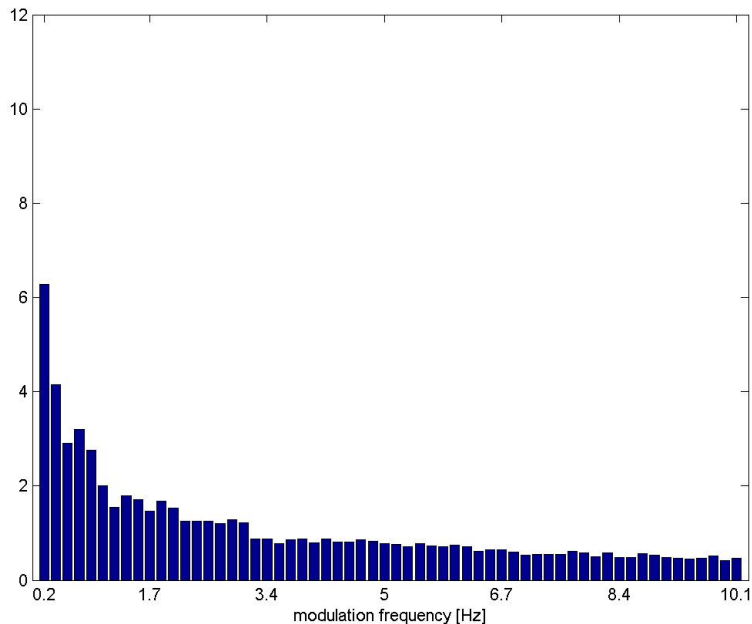


Figure 3

Combination of musical features

In the research of Lidy and Rauber (2005) features are compared and combined to determine their usefulness in setting audio documents apart. This is done by measuring the accuracy in determining the genre for several musical pieces using a combination of features. The result of this research is that the best accuracy is achieved using the features SSD + RH and RP + SSD + RH.

Reliably finding musical features and clustering them is a research area that is still in active development and the current tools are far from perfect.

5. Clustering

For clustering based on features and tags we used the same basic algorithm: K-means clustering. Given a value for k and a set of points (in our case representing tracks), this algorithm divides the set of points in k clusters. Its method is to start with the coordinates of k random points as initial cluster centers. Then for every point in the set it is checked what cluster is nearest and the point is assigned to that cluster. After all the points have been assigned to a cluster, new cluster centers are calculated according to the points assigned to each cluster and the process of assigning points to clusters is repeated until a set number of iterations have been performed or until no (or little) change in the clusters is seen from the one clustering to the next. In the following sections we will give the some of the details of the algorithm when cluster based on features versus when we cluster based on tags and finally how we automatically determine what would be descriptive tags for the entire cluster.

5.1. Feature based clustering

In this topic, the choice for a certain clustering algorithm is discussed. Furthermore, the implementation of this algorithm is also defined.

The first detail to be filled in is how the distance between two points will be determined. In the case of the music features we had three different features that had to factor into the distance. The different values of all three features have an ordinal ordering (ordinal 1dimensional histogram (RH), ordinal 2d histogram (RP) and a similar structure for SSD), meaning adjacent values are more similar in nature than those further away. When measuring distance between ordinal histograms it is undesirable to treat the histogram as a fixed-dimensional vector since this does not take into account the ordering of the values. Each value corresponds to a spectrum band and thus the distances between histograms [1,0,0,0], [0,1,0,0] and [0,0,0,1] should not be all equal. We use the distance-ordinal-histogram algorithm by Cha & Sriharib that takes the ordering of values into account.

Werman, Peleg and Rosenfeld describe a method to unfold multidimensional histograms so that conventional distance metrics can be used on pairs of the two unfolded sets. Because the details of unfolding are not explained in the paper and they conclude it is computationally expensive and may not result in improvement, we did not implement it. Since we were unable to find any other multidimensional (extensions to) histogram distance, we choose to implement it as the (computationally cheap) Euclidian distance between two large vectors.

We thus use ordinal-histogram distance for RH and the seven metrics of SSD and Euclidian distance for RP. We are then still left with the problem that we have 9 (seven from SSD) different distance values that need to be combined into one distance value. Simply adding them up will not do as this will put a larger weight on whichever of those distances happens to be relatively large in general. To solve this problem we calculated the standard deviation of each of the distances over our dataset and normalize the distances, so that SSD, RP and RH are all of equal influence on the final distance used. Finally we also need to calculate the center of clusters according to the points in the cluster. We decided to take the simple approach of taking the average of value of the points as the center.

5.2. Tag based clustering

Cattuto et al. (2008) proposed that the main similarity measures for tags are co-occurrence, cosine similarity and FolkRank. We have chosen to use the traditional clustering algorithm K-means with the similarity measure of cosine similarity, because it yields more synonyms. With the relative scarcity of tags in our track database, this is a great advantage.

Cosine similarity was used to determine the distance between points. For this method every point is seen as a vector where every dimension represents a tag and its value represents how strongly that tag is associated with that point (i.e. track). The cosine similarity looks at the cosine of the angle between the two vectors and returns this as the similarity measure. The advantage of this method is that only the direction of the vector matters and not the size. Note that the fact that we now work with similarity rather than distance requires some changes in the code (a point is grouped with the most similar cluster, rather than to the least distant cluster.)

Updating the cluster centers is done by taking the normalized average of all the points for every tag. This is done because otherwise tracks with many tags will be dominant in determining the new cluster center.

The scarcity problem plays a big role in clustering. Especially in the initial run (when the cluster center is equal to a random point) a lot of similarities will end up as zero. Still assigning these points to a random cluster (or always to the same cluster) is not satisfactory and thus we introduced a garbage cluster. Points that are not similar to any cluster at first will be placed in this cluster and will be clustered again normally in the next iteration. In the end the garbage cluster contains those points that do not fit with any of the clusters created. We found 6 tracks that consistently ended up in the garbage cluster and could thus not be grouped with the other tracks.

5.3. Tagging the cluster

A cluster is unlabeled and only consists of its contents. In order to give an indication of the contents of a cluster we decided to apply an algorithm that looks for the tags that are most applicable for that cluster. The idea is based on TF-IDF. The formula calculates the applicability of a tag for a cluster as follows: $\text{applicability} = (\text{TF} \times \log(\text{ADF}/\text{TDF}))$. Here TF is how often the given tag is relatively found in the given cluster (normalized by the times a track is tagged for every tag). ADF is the same measure, but then over all tag and all tracks. TDF is the same measure for the given tag, but then over all tracks. The idea behind this is that we are looking for tags that are common in the cluster (the TF part), but that are uncommon in the entire dataset (the second part). It thus strikes a balance in choosing tags that are common in the cluster, but not too common throughout the entire dataset.

6. Results

In this topic, the cluster diagrams of both the tag clusters and the musical feature clusters are shown.

First, the similarity between different runs of the tag based clustering is compared. This is important because this proves the stability of the clustering algorithm. If the algorithm is not stable, a comparison of tag and feature based clustering is not meaningful.

The diagram below illustrates the cluster similarity. For each combination of two clusters, the percentage of similarity of tags is shown. As the diagram in Figure 4 below shows, the clusters are not exactly alike. However, the clusters from the first run are often represented by two smaller ones in the second run or vice versa. It seems that the clustering algorithm does show significant variance but this is not an issue for the experiment objective.

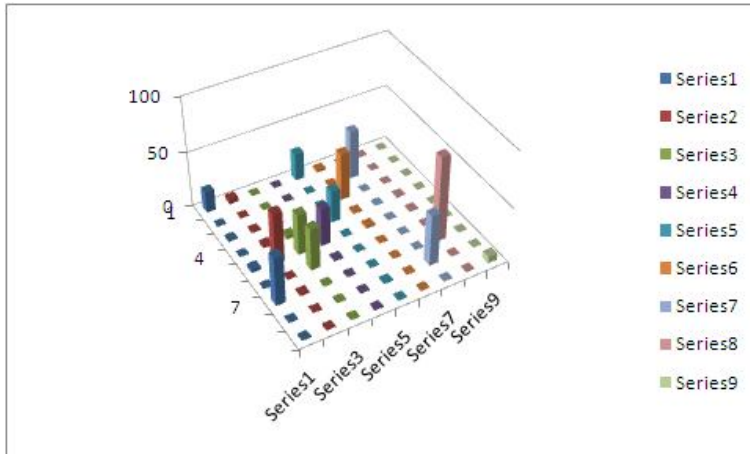


Figure 4

The similarity of two feature clustering runs shows a more consistent result with more clusters having near/over 100% similarity. (Figure 5):

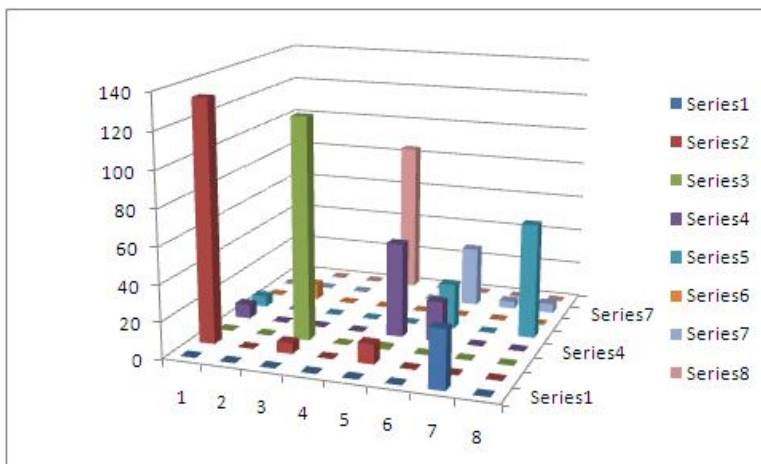


Figure 5

The most important result is the similarity of music feature clusters with the tag based clusters. Judging by the number of bars (and the different scale cutoff), the comparison is lacking similarity. It is obvious that some clusters from the tag based run have tracks spread over almost all of the feature based clusters and vice versa. This implies a low similarity. (Figure 6):

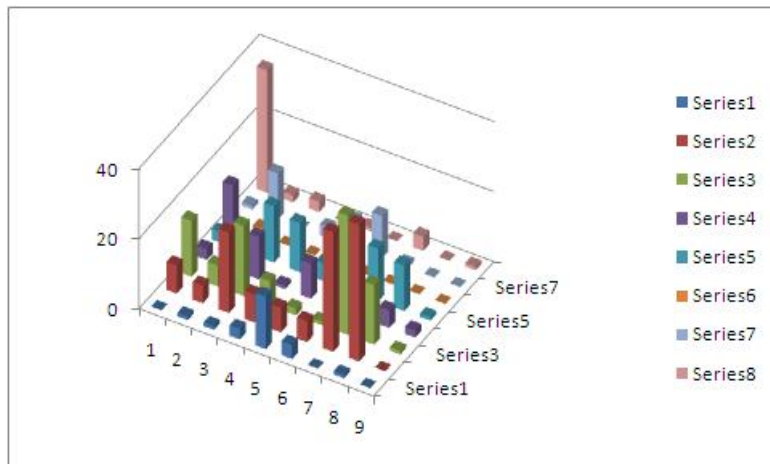


Figure 6

7. Conclusion

From the results it has become clear that the cluster structure between tag based clustering and feature based clustering lacks similarity. The top tags do give a good indication of most prevalent genre in a cluster. Based on these results, the hypotheses are evaluated. The first hypothesis stated:

'By utilizing musical features, it is possible to cluster music by similarity to form a genre'

This hypothesis was not confirmed as there is low similarity of feature based clusters to the tag based clusters, while the theory stated that tag based clusters strongly resemble genre's. An explanation for this observation is that clustering based on features is based on other things than genre. For example, the feature clustering algorithm will consider a soft music track by a rock band to be classical. A user will likely tag this music track as rock regardless, due to the expectation of the artist to produce this kind of genre (genre bias). This shows potential of the feature clustering algorithm: it can avoid this bias and suggest music to the user that is truly alike regardless of the artist.

The second hypothesis stated:

'Clustering by musical features results in a similar network as clustering by social web meta-data'

The results show that the clusterings are not very similar. As stated before, the clustering based on tags is more according to musical genre and the clustering based on musical features is based on the sound alone and not on preconceptions about artists. Another possible explanation is lyric content that might evoke an emotion or feeling from the user and influence the tag.

Furthermore, clusters based on tags are more inconsistent. Clustering based on features becomes worse as number of clusters increases. Music becomes more clustered on some other salient features rather than the general structure.

Lastly, it was confirmed that tag ambiguity and a lack of tags describing musical content do indeed attribute to the problems associated with clustering by tags derived from Last.fm.

8. Discussion

A possible solution for tag ambiguity is the use of MOAT. MOAT is a lightweight Semantic Web framework that provides a way to let content producers give machine readable meaning to tags. (Passant, 2008) This could be used to improve the ambiguity of tags on Last.fm resulting in more reliable clustering. Another solution is the use of latent semantic indexing.

The problem that it is impossible to distinguish between more popular concepts and more general concepts is still present with the use of system clustering. For example, clusters might be named according to the most occurring tag like 'pop' but this is a very general genre, and a more appropriate genre is down the list. A possible solution for this is user specified relations. (Plangprasopchok, Lerman, 2009) However, this was not the focus of this research.

The difficulty of obtaining tags is a problem due to the presence of multiple entries per track on Last.fm. The Last.fm API accepts a Musicbrainz ID and determining this ID solves the identification of the track in question but not the problem of multiple existing entries per track on Last.fm. This results in tags being distributed over multiple tracks and this makes it unclear which track to select when crawling the API.

The lack of tags on Last.fm remains a problem and is therefore not the best resource to cluster music by. The lack of tags is explained by the fact that Last.fm itself uses only the 'scrobbling' data to generate similar music recommendations (question 6 of [13]). Therefore, tags are not the main focus of the site and users do not regard them as such either.

Feature clustering can certainly be improved by looking at better weights for musical features and applying more advanced distance measures. Research in genre detection may give good information on which features are more important to determine genre (and thus more significant in what humans experience as different). Research in this area is very lively which implies there is potential for improvement. We have only touched the surface of Music Information Retrieval as the application of MIR in a Semantic Web context was the main goal of this research.

A clear advantage of feature clustering is however that there is no cold start problem. A track can be suggested as similarity without being evaluated by a user. This avoids the problem where a track new to the system never gains attention as it is never suggested.

9. Applications and future work

A possible application of this research is a web service where users submit the raw feature data of songs (identified by Musicbrainz). This feature data is stored in a openly accessible database. This database can then be used by third party applications to generate recommendations of similar music but there are also other possibilities. Think of an evaluation of your musical taste, music suggestions for a certain mood or handling imperfect matches for Musicbrainz. This is a combination of web services that illustrates the potential of a Semantic Web application.

The essence of similarity by musical features is that it is not limited by the existing categorization (taxonomy) of genres. It can be leveraged to create a music recommendation system that is able to think outside the box.

The extracted features and tags as well as the calculated clusters and scripts used are available at <http://www.rchu.nl/files/HumanVsMachineClusteringMusic.tar.bz2> .

10. Literature

1. Thomas Lidy, Andreas Rauber. Evaluation of Feature Extractors and Psycho-acoustic Transformations for Music Genre Classification. Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005), pp. 34-41, 2005.
2. Passant, A. and Laublet, P., Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data. Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008), Beijing, China, 2008
3. Monachesi, P. and Markus, T., Using social media for ontology enrichment. Utrecht University, Utrecht. 2010
4. Specia, L. and Motta, E., Integrating Folksonomies with the Semantic Web. 4th European Web Semantic Conference – ESWC – 2007. Innsbruck, June 3 – 7, 2007
5. Plangprasopchok, A. and Lerman, K., Constructing Folksonomies from User-Specified Relations. USC Information Sciences Institute. 2009
6. Cattuto, C., Benz, D., Hotho, A., Stumme, G., Semantic grounding of tag relatedness in social bookmarking systems. Proceedings ISWC 2008. LNCS, Karlsruhe, Germany, 2008
7. Levy, M., Sandler, M., A semantic space for music derived from social tags. Centre for Digital Music, Queen Mary, University of London. 2007
8. Michael Werman, Shmuel Peleg, Azriel Rosenfeld, A distance metric for multidimensional histograms, University of Maryland, 1984

9. Sung-Hyuk Chaa, Sargur N. Sriharib. On measuring the distance between histograms. Pattern Recognition Society. Published by Elsevier Science Ltd. 2002

10. Audio Feature Extraction. MIR Group, Vienna Institute of Technology:
<http://www.ifs.tuwien.ac.at/mir/audiofeatureextraction.html>

11. Last.fm Web Services: <http://www.last.fm/api>

12. python-lastfm. A python interface to the Last.fm web services API:
<http://code.google.com/p/python-lastfm/>

13. Econsultancy. Interview with Martin Stiksel of last.fm:
<http://econsultancy.com/blog/485-interview-with-martin-stiksel-of-last-fm>