Musical Similarity Analysis based on Chroma Features and Text Retrieval Methods

David Englmeier[1], *Nina Hubig*[2], *Sebastian Goebl*[1], *Christian Böhm*[1]

1 - University of Munich, 2 - Helmholtz Center Munich
LFE Datenbanksysteme, iKDD-group
{englmeier, hubig, goebl, boehm}@dbs.ifi.lmu.de

Abstract: At the present day the world wide web is full of music. Highly effective algorithms for music compression and high data storage has made it easy to access all kind of music easily. However, it is not possible to look for a similar piece of music or a sound as easily as to google for a similar kind of text. Music is filtered by its title or artist. Although musicians can publish their compositions in a second, they will only be found by high youtube ratings or by market basket analysis. Less known artists need much luck to get heard, although their music might just be what people want to hear. To approach this issue, we propose a new framework called MIRA (Music Information Retrieval Application) for analyzing audio files with existing Information Retrieval (IR) methods. Text retrieval has already yielded many highly efficient and generally accepted methods to assess the semantic distance of different text. We use these methods by translating music into equivalent audio words based on chroma features. We show that our framework can easily match music interpreted even by different artists.

1 Introduction

A modern, effective and personalized music recommendation system needs to focus on multimodal features: musical content and context as well as user correlation [GMS12]. Musical content comprises low-level features acquired by signal processing methods as there are Mel Frequency Cepstral Coefficients or chromatograms, both based on fourier analysis. Others are rate of zeros crossings, upper envelope for loudness, bandwidth analysis or spectral centroids. These low-level features might appear unrelated to music recommendation or melody, but are simple to calculate and, indeed, build a solid base for more sophisticated similarity measures. Mid-level features for musical content apply statistic and machine learning methods on low-level features to gain rythmic information by beat histograms or tonal information as basic melodies or chords. Chroma features are used very frequently. High-level features intend to build user related concepts out of low- and mid-level features. Examples are euphony, infered atmosphere, musical genre or even automatically triggered dance moves. To gain these features, as for text retrieval, semantic features need to be extracted. Additional from physical sound data we need to focus on processes of music perception and reactions to it by various consumers. Insights from physics and psychology, acoustics and cognitive musicology need to be combined to step ahead in music analysis: relations beween sounds and sympathy have to be understood to connect musical content

and recommendation.

Features of musical context comprise information which can not be extracted by music itself. Examples are: meaning of vocals, musician's political background, geographic origin [Dow09]. These features have an influence on perception of a piece of music and, thus, on music recommendation.

User context is also used for music recommendation. It takes search and music listening habits into account. An example for visual aspects are album covers. Music marketing tries to focus covers on potential buyers, although there is no direct connection betweeen music and album art. Choice of music is also a social habit as listeners listen to friends' recommendation. Even larger social groups used musical genres for defining their identity.

For an optimal music recommendation system all three types of features have to be taken into account.

For solving such questions, this paper introduces the framework MIRA. It applies different existing IR-methods, specifically the new Explicit Semantic Analysis (ESA) method to music files. As we focus on chroma features MIRA finds and ranks similar cover versions of a given song title.

The remainder of this paper is the following: First we give some background knowledge on the most interesting applied IR methods in MIRA, Section 3 then shows how these methods are used in the context of music files. The experiment Section 4 shows some results for these methods. Section 5 concludes this paper.

2 Background

This section gives the necessary background information of the most important (1) methods how to extract features from audio files, specifically how to get a chroma feature and (2) to the two most important applied text-retrieval methods for these chroma features.

2.1 Basic methods of feature extraction

The technique in extraction features from audio files strongly determines how the frequency looks like. We introduce the Constant Q Transform and the chroma features that both evolved from the Fast Fourier Transform (FFT).

2.1.1 Constant Q Transform

Constant Q Transform introduced 1990 by Judith C. Brown [Bro90] solves a problem that couldn't be solved by the original FFT algorithm where it bases on: adapting to musical frequencies. The reason is that musical frequencies have no equidistant intervals that are necessary for FFT but act more like links in geometric sequences, in which the quotient of

two closely related links is constant. The interval of two links f_i and f_{i+1} with $f_i = f_0 \cdot 2^{\frac{i}{12}}$ is calculated as follows:

$$\delta_i = f_{i+1} - f_i = f_i(2^{\frac{1}{12}} - 1)$$

As can be seen, the interval depends on the frequence f_i and is not the same for all pairs of frequencies as is the case for DFT.

2.1.2 Chroma features

But even the wide tone spectrum of the Constant Q Transform is due to the large number values not entirely able to be a similarity measure for music pieces. Its values have no meaning for a single music piece. Thus, it is further edited to gain another significant recognition feature: the chroma feature. As found by [GMS12] the chromagram created from the chroma features represents an audio feature that reflects the characteristics of the music piece well and is on the other hand immune to specific attributes of recordings of the same song. To create such chroma vectors all tones of different octave of the corresponding 12 half tones are mapped into one octave. This means that for example tone "A" is added to a value, whose sum represents a component of the chroma vector, regardless of its respective octave. This is regarded as some sort of weight and can be calculated as follows [Lug09]:

$$CV(i) = \sum_{m=0}^{M-1} |X_{CQ}(i + 12m)|$$

where i is the entry index of a respective chroma vector CV, M is the number of octaves and X_{CQ} is the vector calculated by the Constant Q Transform. The standard way to enhance the robustness of a chroma vector is normalization. It makes the CV specifically robust to changes in volume and dynamic of the audio file [GMS12]. These positive effects through postprocessing as well as the robustness make chroma vectors a sensible choice for finding live - or cover-versions of the same song [RHG08]. In Figure 1 you see a chromagram that shows in a time line the different exististing chroma vectors of the classical music piece "For Elise" by Ludwig van Beethoven. The intensity of the tones are colored in either red (high amplitude), green (medium amplitude) to blue (low amplitude).

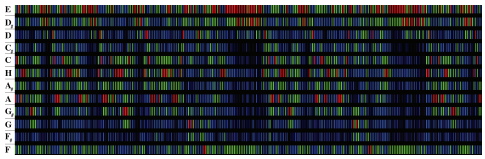

Figure 1: Chromagram: Ludwig van Beethoven, title: „For Elise“

2.2 Applied IR techniques

Here we explain the two most important text- or information retrieval techniques used in this framework.

2.2.1 TF-IDF weights

The TF-IDF weights is a measure which expresses the meaning of a term or a document within a collection of documents. This measure was established in IR by Salton and Yang [SY73] and will be used in this framework on audio words, terms and songs as documents in the respective song collection. The TF-IDF measure is composed of two weights: a local weight depending on the document and a global weight that takes the complete document collection into account.

The local weight $tf_{i,j}$ is based on the frequency of a given term i in a document j and calculated by the following formula [BYRN11]:

$$tf_{i,j} = \frac{h_{i,j}}{\sum_k h_{k,j}}$$

The global weight uses the document frequency df_i of a term i, which defines the number of documents the term exists. It is irrelevant for the document frequency how often a term occurs in a single document as calculated in term frequency tf. The formula for the df is given according to [Jon04]:

$$df_i = \frac{|d_k : t_i \in d_k|}{|D|}$$

In information retrieval usually the inverse document frequency idf is calculated, due to the semantics that terms that occur in all documents should not have a specific meaning for a single document:

$$idf_i = \log \frac{|D|}{df_i}$$

Consisting of the local term frequency tf and the global (inverse) document frequency idf the TF-IDF weights are created:

$$(tfidf)_{i,j} = \frac{h_{i,j}}{\sum_k h_{k,j}} \times \log \frac{|D|}{df_i}$$

2.2.2 Explicit Semantic Analysis

Explicit Semantic Analysis (ESA) is a relative new method from 2007, that was originally developed for applying it to the wikipedia encyclopedia for finding semantic similarity

between documents [GM07]. The idea behind this method is that a wikipedia article in principle only exists around one topic, a so called concept. As lots of concepts exists in the huge amount of documents of wikipedia, the TF-IDF weights are used for all terms of an article. This results in the term-document matrix, where the columns represent an wikipedia article and the rows represent the terms. The semantic meaning, or concept, correlates to a complete row in the term-document matrix, where the TF-IDF is applied to calculate its relevance. After that, all center vectors of the terms are calculated to know the concepts related to one specific term. With these document vectors a comparison is possible by applying for example the cosine similarity [TP07]. Overall the function of ESA makes it possible to find for every given term and/or document a representation, that considers the relevance of the term or document to all existing concepts.

3 MIRA Framework

In this section we present our graphical user interface (GUI-) application MIRA (Music Information Retrieval Application) that measures similiarity between musical works by applying different IR-methods on chroma features. Specifically in this section we focus on the novel ESA evaluation MIRA is handling as first framework in this area. We start with how we create the audio words in MIRA:

3.1 Creating Audio words from music

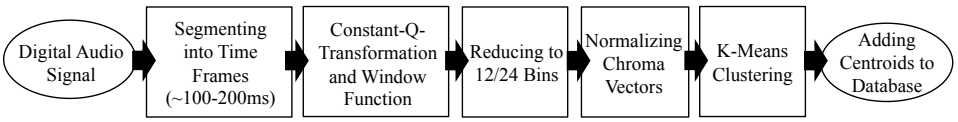

Figure 2: Workflow for creating audio words.

The first step for evaluating musical similarity is creating the needed audio words from the files. The workflow for this implemented in MIRA is outlined in Figure 2. As segmentation, normalization and k-means clustering works straightforward we want to explain the used constant q transform and window function in a bit more detail:

Efficient Constant Q Transform MIRA uses a combined implementation of the Constant Q Transform by Blankertz [Bla01] combined with a Java Implementation from Karl Helgason from 2006. Blankertz optimizes the Constant Q Transform in his algorithm by efficiently multiplying the temporal kernel matrix T into spectral space as suggested in BLA01. This is done by applying FFT on the columns of T. So instead of calculating with $\vec{c} = \vec{x} \cdot T^*$ we calculate efficiently $\vec{c} = \frac{1}{N} x^{ft} \cdot S^*$. Thus S^* is a sparse Matrix with lots of zero values we need much less matrix multiplications than before.

Efficient Window Function As suggested by Blankertz and Brown [Bla01] MIRA uses a Hamming window for balancing frequency errors that happen by randomly chosing time fragments of songs. Such a window function with $\alpha = 0.54,\ \beta = 1 - \alpha = 0.46.$ is given by:

$$w(n) = \alpha - \beta\ cos\left(\frac{2\pi n}{N-1}\right)$$

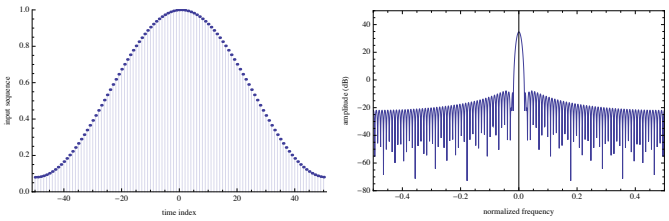

Figure 3: The Hamming window (left) is smoothening the signal. Right we see an application of this window to a signal in the fourier transform.

Figure 3 shows a signal sent without window function - a so called rectangular window on the left and a Hamming window on the right. Clearly, the Hamming window strongly reduces bins with high amplitude, which allows it easier to recognize single frequencies.

3.2 Evaluation of Audio Words

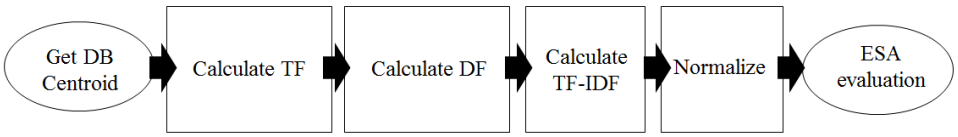

Figure 4: Workflow for evaluating audio words.

As the workflow of MIRA for evaluating audio words is quite intuitive, some considerations need to be done before applying the given IR methods to music:

Applying TF-IDF weights to music As TF-IDF is a method that originally tries to semantically analyze texts, the transfer to music is not easily possible. First note that through clustering created audio words in general have no semantic meaning on their own. Only through TF-IDF a semantic information can be given to every audio word because highly redundant audio words like repeating musical elements are getting a smaller relevance overall. Besides, chromatic information gives no further information on genre or

interpret of a given song. How exactly can we measure the results then? TF-IDF on audio words is useful for recognizing the correct cover versions.

Applying ESA to music For applying ESA to music we analogously to Section 2.2.2 characterize every audio regarding its relevance compared to all given songs of the database. This relevance is calculated by the TF-IDF as described in Paragraph 3.2 and depending on the center of all audio word vectors. As ESA is a method used for very large data sets, our data base of around 4000 songs could be a drawback.

4 Experiments

The experiments done for this paper are two-fold: First, we want to show the quality of the extracted chroma features for finding similarity between cover versions of a song to the original. Second, we apply and evaluate ESA on our database for optimizing the ranking result of similar audio files.

4.1 The quality of chroma features

Lets take a look how effective different chroma features can be differentiated for different interpretations and tonalities in the audio files:

Classical Music Example The highest quality and the highest musical similarity with chroma features on different interpreted audio files can be achieved with classical music. Here in Figure 5 can be even visually, with human eye, shown that chroma features give a solid basis for musical similarity.

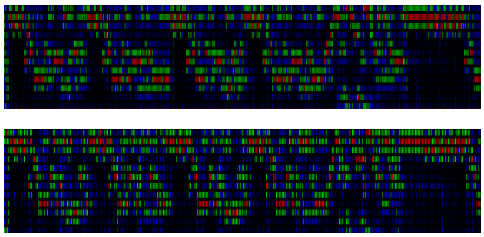

Figure 5: Comparison: Ludwig van Beethoven (original), title „For Elise“, interpreted by Sylvia Capova (upper) und Ivo Pogorelich (lower)

Same tonality but polyphonic voices Although given the same tonality as in the classical music example but with quite different stylistic cover especially with many polyphonic

voices, the song has very little similarities as can be seen in Figure 6.

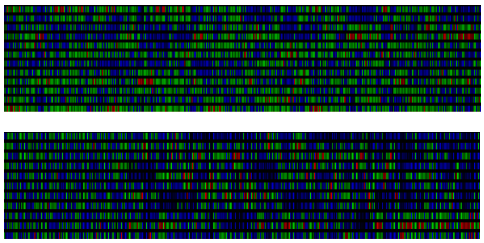

Figure 6: Comparison: Lykke Li „I Follow Rivers“, interpreted by Lykke Li (upper) and Triggerfinger (lower)

Different tonalities The last comparison shows two songs, that are stylistically similar but interpreted in different tonalities. Figure 7 shows that the chromagram does indeed show similar pattern, but the displacement of the chroma vector components makes a mapping of identical audio words without correction impossible.

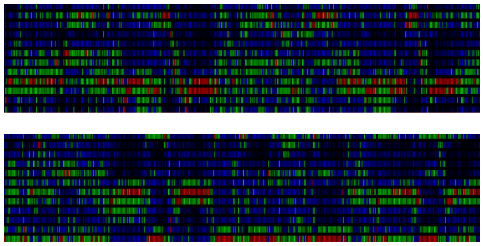

Figure 7: Comparison: Hank Williams „Ramblin’ Man“, interpreted by Hank Williams (upper) and Del Shannon (lower)

4.1.1 Explicit Semantic Analysis Evaluation

The following evaluation refers to a given database with around 4000 songs, as described in a specification by Riley, Heinen and Ghosh [RHG08] as well as Constant-Q-parameter settings described by Brown [BP92]. By applying a standard evaluation within a feature space applying the cosine similarity as done in the above mentioned paper, some cover versions are not recognized accurately. To improve the ranking we newly want to test the explicit semantic analysis (ESA). As ESA has high computational costs we only apply it to already falsely ranked results.

Ranking with ESA For testing the ranking of ESA we chose two songs, which were as live covers of „Wicked Game“ from Chris Isaak as well as „Bad Moon Rising“ from Creedence Clearwater Revival falsely ranked compared to their studio version.

Rang	Song 1	Song 2	Winkel
1	*Wicked Game* Chris Isaak	*Wicked Game* Chris Isaak (live)	62,0°
21	*Wicked Game* Chris Isaak	*Wicked Game* Chris Isaak (live)	46,9°

Figure 8: Comparison of the Chris Isaak song „Wicked Game“. Red: Bad ranking without ESA. Green: Correct ranking with ESA. ”Winkel” refers to the cosine similarity

Rang	Song 1	Song 2	Winkel
7	*Bad Moon Rising* Creedence Clearwater Revival	*Bad Moon Rising* Creedence Clearwater Revival (live)	71,3°
26	*Bad Moon Rising* Creedence Clearwater Revival	*Bad Moon Rising* Creedence Clearwater Revival (live)	45,3°

Figure 9: Comparison of the Creedence Clearwater Revival song „Bad Moon Rising“. Red: Bad ranking without ESA. Green: Correct ranking with ESA. ”Winkel” refers to the cosine similarity

Our experiment shows for both song examples a much more effective recognition of the cover-versions when ESA is applied. While the cover identification of „Bad Moon Rising“ by Creedence Clearwater Revival was raised from rank 26 to rank 7, has „Wicked Game“ by Chris Isaak even improved to rank 1 - 20 ranks higher than before.

Scalability of ESA Clearly, the ESA evaluation proved useful for improving false ranks, but still the roots of ESA has to be considered: ESA is a text-retrieval method and as such conceptionally prone to work on a much larger amount of terms and documents as is the case here. Although Riley, Heinen and Ghosh claim a number of 500 audio words as the most effective [RHG08] without any proof, we experiment by scaling the number of audio words for ESA. As we have no larger database of songs than 4000 we only scale the number of audio words, especially the k-means cluster centers, from 500 to 10000. By doing so we receive no further positive effect on the ranking against our hypothesis for further improvement.

5 Conclusion

Concluding this paper we showed first, how chroma features are interesting and reliable for differentiating cover versions of music files. Furthermore we exploited how different IR methods like TF-IDF weights and explicit semantic analysis (ESA) are useful for optimizing the ranking in musical similarity analysis. We determined this by vast experiments done with our GUI - application MIRA. We see future work in analyzing different kind of features than chroma vectors to efficiently answer other research questions than how close cover versions are related to the original.

References

[Bla01] Benjamin Blankertz. The constant Q transform, 2001.

[BP92] Judith C. Brown and Miller S Puckette. An efficient algorithm for the calculation of a constant Q transform. *The Journal of the Acoustical Society of America*, 92:2698, 1992.

[Bro90] Judith C. Brown. *Calculation of a constant Q spectral transform*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1990.

[BYRN11] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, London, 2011.

[Dow09] J Stephen Downie. Music Information Retrieval Meets Education. *The Music Information Retrieval Evaluation Exchange MIREX*, pages 247–255, 2009.

[GM07] Evgeniy Gabrilovich and Shaul Markovitch. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

[GMS12] Peter Grosche, Meinard Mller, and Joan Serra. Audio Content–Based Music Retrieval. *Multimodal Music Processing*, pages 157–174, 2012.

[Jon04] Karen Sp rck Jones. IDF term weighting and IR research lessons. *Journal of Documentation*, 60(5):521–523, 2004.

[Lug09] Michael Lugmair. *Ein Dynamic Time Warping basiertes Score Following System*. PhD thesis, Universitt Augsburg, 2009.

[RHG08] Matthew Riley, Eric Heinen, and Joydeep Ghosh. A Text Retrieval Approach to Content-Based Audio Retrieval. *Ismir 2008, Session 3a, Content-Based Retrieval, Categorization and Similarity I*, pages 295–300, 2008.

[SY73] Gerard Salton and Chung-Shu Yang. On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372, 1973.

[TP07] Sandeep Tata and Jignesh M Patel. Estimating the selectivity of tf-idf based cosine similarity predicates. *ACM SIGMOD Record*, 36(2):7–12, 2007.