Feature Extraction for Music Information Retrieval

Jensen, Jesper Højvang

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

Feature Extraction for Music Information Retrieval

Ph.D. Thesis

Jesper Højvang Jensen

Multimedia Information and Signal Processing
Department of Electronic Systems
Aalborg University
Niels Jernes Vej 12, 9220 Aalborg Ø, Denmark

Feature Extraction for Music Information Retrieval
Ph.D. thesis

ISBN XX-XXXXX-XX-X
June 2009

Abstract

Music information retrieval is the task of extracting higher level information such as genre, artist or instrumentation from music. This thesis is concerned with music information retrieval in the case where only sampled audio is available, i.e., where higher level information about songs such as scores, lyrics or artist names is unavailable. In the introductory part of the thesis, we give an overview of the field. We first briefly describe the dominating topics and outline the practical as well as the fundamental problems they face. In the last half of the introduction, we give a more detailed view of two specific music information retrieval topics, namely polyphonic timbre recognition and cover song identification.

In the main part of the thesis, we continue with these two topics. In Paper A–C, we consider a popular measure of timbral similarity, which is frequently used for genre classification. In Paper A, we analyze the measure in depth using synthesized music, in Paper B we compare variations of this measure including a version that obeys the triangle inequality, and in Paper C, we compare different mel-frequency cepstral coefficient estimation techniques.

In Paper D and E, we introduce a fast cover song identification algorithm and a representation of rhythmic patterns, respectively, that both utilize compact features that are insensitive to tempo changes. In Paper F, we evaluate a number of features commonly used in music information retrieval.

In Paper G, we derive the maximum likelihood joint fundamental frequency and noise covariance matrix estimator, and finally, in Paper H, we analyze two different approaches to fundamental frequency estimation using optimal filtering.

List of Papers

The main body of this thesis consists of the following papers:

[A] J. H. Jensen, M. G. Christensen, D. P. W. Ellis, S. H. Jensen, "Quantitative analysis of a common audio similarity measure", in *IEEE Trans. Audio, Speech, and Language Processing*, vol. 17(4), pp. 693–703, May 2009.

[B] J. H. Jensen, D. P. W. Ellis, M. G. Christensen, S. H. Jensen, "Evaluation of distance measures between Gaussian mixture models of MFCCs", in *Proc. International Conf. on Music Information Retrieval*, 2007, pp. 107–108.

[C] J. H. Jensen, M. G. Christensen, M. Murthi, S. H. Jensen, "Evaluation of MFCC estimation techniques for music similarity", in *Proc. European Signal Processing Conference*, 2006, pp. 926–930.

[D] J. H. Jensen, M. G. Christensen, D. P. W. Ellis, S. H. Jensen, "A tempo-insensitive distance measure for cover song identification based on chroma features", in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, 2008, pp. 2209–2212.

[E] J. H. Jensen, M. G. Christensen, S. H. Jensen, "A tempo-insensitive representation of rhythmic patterns", in *Proc. European Signal Processing Conference*, 2009.

[F] J. H. Jensen, M. G. Christensen, S. H. Jensen, "A framework for analysis of music similarity measures", in *Proc. European Signal Processing Conference*, 2007.

[G] J. H. Jensen, M. G. Christensen, S. H. Jensen, "An amplitude and covariance matrix estimator for signals in colored gaussian noise", in *Proc. European Signal Processing Conference*, 2009.

[H] M. G. Christensen, J. H. Jensen, A. Jakobsson and S. H. Jensen, "On optimal filter designs for fundamental frequency estimation", in *IEEE Signal Processing Letters*, vol. 15, pp. 745-748, 2008.

The following additional papers have been published by the author during the Ph.D. studies:

[1] M. G. Christensen, J. H. Jensen, A. Jakobsson and S. H. Jensen, "Joint Fundamental Frequency and Order Estimation using Optimal Filtering", in *Proc. European Signal Processing Conference*, 2009.

[2] J. H. Jensen, M. G. Christensen, S. H. Jensen, "A chroma-based tempo-insensitive distance measure for cover song identification using the 2D auto-correlation function", in *MIREX Audio Cover Song Contest system description*, 2008.

[3] J. H. Jensen, D. P. W. Ellis, M. G. Christensen, S. H. Jensen, "A chroma-based tempo-insensitive distance measure for cover song identification", in *MIREX Audio Cover Song Contest system description*, 2007.

Besides the above papers, the author has been the lead developer behind the Intelligent Sound Processing MATLAB toolbox[1]. Most of the code used in the papers is available as part of this toolbox.

Fig. 1: The logo of the Intelligent Sound Processing MATLAB toolbox.

[1] `http://isound.es.aau.dk/`

Preface

This thesis is written in partial fulfillment of the requirements for a Ph.D. degree from Aalborg University. The majority of the work was carried out from August 2005 to October 2008 when I was a full-time Ph.D. student at Aalborg University. However, the papers were not joined with the introduction to form a complete thesis until spring 2009, since I got a position as postdoctoral researcher within the field of speech synthesis in November 2008, thus leaving thesis writing for vacations and weekends. The Ph.D. was supported by the Intelligent Sound project, Danish Technical Research Council grant no. 26–04–0092.

I would like to thank my two supervisors, Mads Græsbøll Christensen and Søren Holdt Jensen, for their advice, encouragement and patience, and for giving me the opportunity to pursue a Ph.D. I would also like to thank all my colleagues at Aalborg University in the section for Multimedia, Information and Speech Processing and in the former section for Digital Communications for fruitful discussions and for making it a great experience to be a Ph.D. student. I am also very grateful to Dan Ellis and his students in LabROSA at Columbia University who welcomed me in their lab in spring 2007. My encounter with New York would not have been the same without you. I would also like to thank all my colleagues in the Intelligent Sound project at the Department of Computer Science and at the Technical University of Denmark.

Last but not least, I thank my family and friends for encouragement and support.

Contents

Introduction

1 Music Information Retrieval

Digital music collections are growing ever larger, and even portable devices can store several thousand songs. As Berenzweig humorously noted in [1], the capacities of mass storage devices are growing at a much higher rate than the amount of music, so in ten years time, a standard personal computer should be able to store all the music in the world. Already today, cell phone plans with free access to millions of songs from the Big Four (EMI, Sony BMG, Universal Music and Warner Music Group) as well as numerous smaller record companies are available on the Danish market[2]. Accessing large music collections is thus easier than ever, but this introduces a problem that consumers have not faced to this extent before: how to find a few interesting songs among the millions available.

As a consequence of the easy access to music, the field of music information retrieval (MIR) has emerged. This multifaceted field encompasses many different areas including, but not limited to, archiving and cataloging, audio signal processing, database research, human-computer interaction, musicology, perception, psychology and sociology. In this thesis we are only concerned with methods to automatically analyze, browse, recommend or organize music collections for end-users. We thus only consider sampled audio and not symbolic audio such as scores or midi files, since transcriptions are not generally available for consumer music collections, and we do not consider the special needs of professional users, such as musicologists or librarians.

Even with this narrower scope, music information retrieval still encompasses many different fields, such as

- Artist recognition
- Audio fingerprinting
- Audio thumbnailing

[2] `http://musik.tdconline.dk/`

- Chord recognition
- Cover song detection
- Female/male vocal detection
- Genre classification
- Instrument/timbre identification
- Mood classification
- Music recommendation
- Rhythmic similarity
- Song similarity estimation
- Tempo estimation
- Transcription

As we are focusing on the signal processing aspects of music information retrieval, we will not directly address e.g. storage problems or psychological or sociological aspects of music in the main body of the thesis, although we will briefly touch some of these issues in the introduction.

1.1 The Curse of Social Science

Music information retrieval tasks for sampled audio can roughly be divided into two categories (see Table 1), namely objective problems and cultural problems. The objective problems are primarily related to the intrinsic properties of music, such as instrumentation, melody, harmonies and rhythm, and we can unambiguously state whether an algorithm succeeds or not, while for the cultural problems, the cultural context plays a large role, and background information is necessary to determine if a proposed algorithm returns the correct result. As an example of the former, consider instrument identification where the instruments in a song are indisputably given, and we can unambiguously state whether we were able to recognize them or not. On the other hand, if we for instance consider genre classification, one can often argue whether the genre of a song is e.g. pop or rock, and the answer often depends on previous knowledge about the artist. Musical genres are not exclusively defined by musical properties, but also by cultural context such as geographical area, historical period, and musical inspiration [2]. Another example of a non-objective problem is mood classification, where the theme song from MASH, the Korean War field hospital comedy, is a good example of how difficult it can be to assign discrete labels. People that have only heard the acoustic MASH theme song from the TV series probably consider it a merry song. However, when hearing the lyrics that are present in the 1970 movie that lay ground to the series, the mood of the song changes significantly. The refrain goes like this:

Objective	Cultural
Artist recognition	Genre classification
Chord recognition	Mood classification
Cover song detection	Music recommendation
Female/male vocal detection	Rhythmic similarity
Fingerprinting	Song similarity
Instrument/timbre identification	
Tempo estimation	
Transcription	

Table 1: Music information retrieval tasks split into objective and cultural tasks. Since most tasks contain both objective and cultural elements, such a breakdown is inevitably oversimplified.

Suicide is painless,
It brings on many changes,
And I can take or leave it if I please.

After becoming aware of the lyrics, the acoustic TV theme suddenly seems much more melancholic. To add a third dimension, the MASH movie is indeed a comedy, so the lyrics can also be considered ironic.

While context-related problems are not that common in signal processing, they are the norm rather than the exception in social sciences. As such, context-related problems in music information retrieval is a natural consequence of it being a melting pot of different scientific fields, and we will therefore take a short detour from signal processing to take a philosophical look at the limits of automated interpretation.

In his 1991 Dr. Techn. thesis[3], Flyvbjerg argues why social sciences are fundamentally different from natural sciences and should not be treated as such [3] (see [4] for an English translation). In the following we will briefly summarize the argumentation, as it is quite relevant to music information retrieval. The basis of Flyvbjerg's argumentation is a model of the human learning process by Hubert and Stuart Dreyfus that describes five levels of expertise that one goes through from being a beginner to becoming an expert. Ordered by the level of expertise, the five levels are:

Novice: The novice learns a number of context independent rules of action that are blindly followed. When learning to drive, this would e.g. be to change gears at certain speeds. The novice evaluates his/her performance based on how well the rules are followed.

[3]A Danish higher doctorate degree not to be confused with the Ph.D.

Advanced beginner: Through experience, the advanced beginner in a given situation recognizes similarities to previous experiences, and context begins to play a larger role. Rules of action can now be both context dependent and context independent. A driver might e.g. change gears based on both speed and engine sounds.

Competent performer: With more experience, the number of recognizable elements becomes overwhelming, and the competent performer starts to *consciously* organize and prioritize information in order to only focus on elements important to the problem at hand. The competent performer spends much time planning how to prioritize and organize, since this cannot be based on objective rules alone. This also results in *commitment*. The novice or advanced beginner only sees limited responsibility and tends to blame insufficient rules if he/she fails despite following them, while the competent performer sees failure as a consequence of his/her insufficient judgment or wrong prioritization.

Skilled performer: Based on experience, the skilled performer *intuitively* organizes, plans and prioritizes his/her work, with occasional analytic considerations. Planning is no longer a separate step, but happens continuously. The actions of the skilled performer cannot be described by analytic rules, but are instead based on the experience gained from countless similar situations.

Expert: Besides intuitively organizing, planning and prioritizing, the expert also acts holistically; there is no distinction between problem and solution. Flyvbjerg gives the example that pilots still learning to fly reported that "they were controlling the plane", while after becoming experienced "they were flying", hinting at a more holistic view [3].

This model of learning explains that intuition is not a supernatural black art to be avoided, but the result of having experienced countless similar situations before. It is an everyday phenomenon practiced by any skilled performer. It also explains why practical experience is needed within almost any field in order to advance from the beginning levels. Dreyfus and Dreyfus coin the term "arational" to describe the skilled performer's intuitive, experience-based way of thinking that is not rational in the analytic sense, but which is also not irrational in the common, negatively loaded sense [3]. Physicist Niels Bohr would probably have appreciated Dreyfus and Dreyfus' model of learning. He has been quoted for saying that "an expert is a person who has made all the mistakes that can be made in a very narrow field," stressing the importance of practical experience to becoming an expert[4].

[4]He has also been quoted for saying "No, no, you're not thinking; you're just being logical," which also supports the necessity of arational thinking to obtain true expertise. However, the

Although there are many tasks in life where we never reach the higher levels of expertise, Flyvbjerg's point is that most social interaction takes place on the expert level. This is somewhat unfortunate if one treats social sciences in the same way as natural sciences, where one can hypothesize a limited set of rules and verify or discard them by experiments, since only the lowest two or three levels of expertise are based on analytic rules, while most human interaction takes place on the intuitive, context-dependent expert level. We do not follow rigid rules when interacting with other people; our behavior holistically depends on who we are, where we are, who else is there, how we are related to them, when we last met them, career and personal situations, etc. Most human behavior simply cannot be described by a small set of rules, and if we attempt to include the relevant context in the rules, we quickly end up describing the entire context itself, including several life stories. Although the discussion at first might sound pessimistic on behalf of social sciences, Flyvbjerg's intention is not to render them useless, but on the contrary to advocate that social sciences would be much more successful if they were based on case studies, which explicitly do consider context, instead of somewhat awkwardly being forced to fit into the framework of natural sciences.

For music information retrieval, the consequence of this is that since music is a cultural phenomenon as much as a natural science, many properties of a song relate to context and thus can neither be derived from the music itself nor from a limited set of cultural rules. Methods to identify the genre, mood etc. of songs that do not take context, e.g. geographic origin, production year etc., into account, will therefore be fundamentally limited in performance. Ultimately, the user of the algorithms should be taken into account as well. This is not necessarily impossible; companies such as Pandora and last.fm have successfully launched music recommendation services that do include context. Last.fm tracks user listening behavior, and Pandora uses trained musicians to categorize all songs. Lately, Apple has also introduced a function in their iTunes music player that automatically generates a playlist of songs similar to a user-selected seed. This function is also at least partially based on user feedback.

The success of these services that manage to incorporate cultural knowledge, and the lack thereof for algorithms based on acoustic information alone, seems to confirm that since music is a cultural phenomenon, the cultural music information retrieval algorithms of Table 1 will be fundamentally limited in performance if the cultural context is ignored. Of course, the objective music information retrieval tasks by nature do not suffer from these limitations, since all the necessary information is already present in the songs.

The consequence of this is that signal processing algorithms for music information retrieval should focus on the objective measures alone, and leave the

quote was directed to Albert Einstein, whom we can hardly consider a beginner when it comes to physics ...

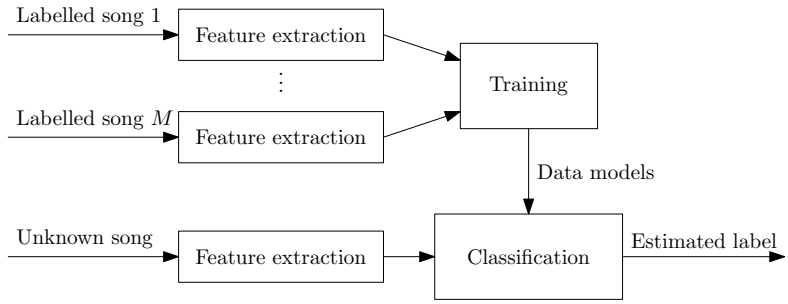


Fig. 2: Block diagram of typical music information retrieval systems with supervised learning.

subjective, interpretative tasks to data mining algorithms that can combine objective properties with cultural knowledge obtained from e.g. training data, user feedback or internet search engines. However, mixing the two, as has been done with genre classification in e.g. [5–10] and by ourselves in Paper C, only serves to blur whether improvements are caused by improved signal representations or improved data mining/classification techniques. Furthermore, such evaluations tend to somewhat artificially impair algorithms by removing relevant information such as artist name and song title.

1.2 MIR systems at a glance

Most music information retrieval systems either analyze a song and designate a discrete label, such as a genre or artist classification system, or return a measure of distance/similarity between two songs, such as is done by many cover song identification systems. Fig. 2 and 3 show a block diagram for each of these scenarious, respectively. Common to both block diagrams is that songs are never used directly for classification or distance computation, but that compact, descriptive features are extracted from the songs as an intermediate step. In Section 2 and 3, examples of such features will be presented, and in Paper F, we have compared the performance of some such features.

Trained systems

For the trained systems in Fig. 2, a set of annotated training songs are used to train a classifier, which for instance could be Gaussian mixture models [10, 11] or support vector machines [12–14]. The trained classifier is now used to predict the labels of unknown songs. The advantage of trained systems is that they usually perform better than untrained systems. The downside of trained systems is that labeled training data is needed, which not only forces the use of a single

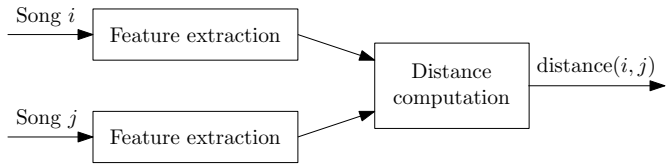


Fig. 3: Block diagram of typical a untrained music information retrieval system. Using the nearest neighbor classifier, this is a special case of Fig. 2.

taxonomy that songs might not always fit into, but labels may also change as e.g. new genres evolve. Annotating songs can be quite labor intensive, even though the number of labels needed can be reduced using active learning techniques [15].

Untrained systems

Untrained systems (see Fig. 3) do not employ a classifier. Instead, the algorithms use the extracted features from two songs to compute a measure of distance (or similarity) between them. With untrained algorithms, it is not necessary to define categories, e.g. genres, a priori. This makes it possible to give a song as seed and retrieve the songs most similar to it, for instance for use in a playlist. The most similar songs, i.e., the songs with the shortest distances to the seed, are often called the nearest neighbors. When evaluating untrained systems in the framework of trained systems where labeled training data is available, the k-nearest neighbor algorithm is commonly used. The k nearest neighbors in the training data to a seed with unknown label are retrieved, and the seed is assigned the most frequent label among the k nearest neighbors. As the amount of data approaches infinity, the k-nearest neighbor algorithm is guaranteed to approach Bayes error rate, which is the minimum error rate given the distribution, for some k. The nearest neighbor algorithm, which is the special case where $k = 1$, is guaranteed to have an error rate no worse than two times Bayes error rate [16]. The nearest neighbor (or k nearest neighbor) classifier is thus a good choice when the classifier is only of secondary interest. In our experiments in Paper A to F, we have used the nearest neighbor classifier.

The triangle inequality

If the distances between songs returned by a system obey the triangle inequality, then for any songs s_a, s_x and s_c, and the distances between them, $d(s_a, s_c)$, $d(s_a, s_x)$ and $d(s_x, s_c)$, the following holds:

$$d(s_a, s_c) \leq d(s_a, s_x) + d(s_x, s_c). \tag{1}$$

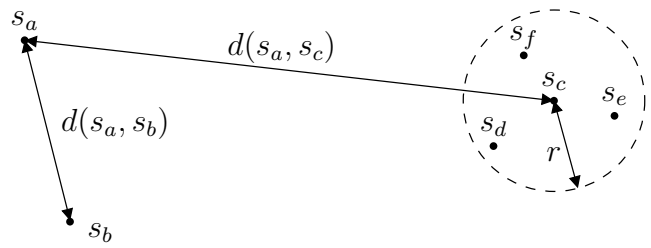


Fig. 4: Sketch showing how to use the triangle inequality to find the nearest neighbor efficiently. The song s_a is the seed, and s_b is a candidate to the nearest neighbor with distance $d(s_a, s_b)$. The song s_c is the center of a cluster of songs, where any song s_x in the cluster are within a distance of r to the center, i.e., $d(s_c, s_x) \leq r$. By the triangle inequality, we see that for any s_x in the cluster, $d(s_a, s_x) \geq d(s_a, s_c) - d(s_c, s_x) \geq d(s_a, s_c) - r$. If, as the drawing suggests, $d(s_a, s_c) - r > d(s_a, s_b)$, we can without additional computations conclude that no song in the cluster is closer to s_a than s_b.

In words, this means that if songs s_a and s_x are similar, i.e., $d(s_a, s_x)$ is small, and if songs s_x and s_c are similar, then s_a and s_c will also be reasonably similar due to (1). This can be used to limit the number of distances to compute when searching for nearest neighbors. Rearranging (1), we obtain

$$d(s_a, s_x) \geq d(s_a, s_c) - d(s_x, s_c). \tag{2}$$

If we search for the nearest neighbors to s_a, and we have just computed $d(s_a, s_c)$ and already know $d(s_x, s_c)$, then we can use (2) to bound the value of $d(s_a, s_x)$ without explicitly computing it. If we already know another candidate that is closer to s_a than the bound, there is no need to compute the exact value of $d(s_a, s_x)$. Hence, we save distance computations, but the price to pay is that we need to precompute and store some distances. This is depicted graphically in Fig. 4. In Paper B, we describe a distance measure between songs that obey the triangle inequality.

Because of the curse of dimensionality, it depends on the intrinsic dimensionality of the distance space how many computations we can actually save by exploiting the triangle inequality. The curse of dimensionality states that as the dimensionality of a vector space increases, the distance between arbitrary vectors in this space approaches a constant in probability.

Several authors have observed that for distance-based audio similarity measures, a few songs sometimes show up to be the nearest neighbor to a disproportionate number of songs without any obvious reason [1, 17]. Although it has not been formally proved, it is expected that this is also linked to the curse of dimensionality [1]. Thus, for untrained MIR algorithms, there are several good reasons that features should be low-dimensional, or at least be embedded in a low-dimensional manifold.

1.3 Obtaining Ground Truth Data

For long it was impractical to compare different music information retrieval algorithms, since copyright issues prevented the sharing of the music collections that could standardize the evaluations. The annual MIREX evaluations, which are held in conjunction with the International Conference on Music Information Retrieval (ISMIR), is an attempt to overcome this problem by having participants submit their algorithms which are then centrally evaluated. This way, distribution of song data is avoided, which also has the advantage that overfitting of algorithms to a particular data set is avoided. The latter advantage should not be neglected, as demonstrated in e.g. [18] and by our 2008 cover song identification algorithm [19]. Our algorithm showed an increase of accuracy from 38% to 48% on the covers80 [20] data set, which was used for development, but in the MIREX evaluation the number of recognized cover songs rose from 762 to 763 of the 3300 possible, an increase of 0.03 percentage points ... Below, we briefly describe the most commonly used music collections.

In-house collections: Many authors, such as [5, 8, 21, 22], have used in-house collections for evaluations. As these collections cannot be legally distributed, it is difficult to compare results directly.

RWC Music Database: The Real World Computing (RWC) Music Database was created by the Real World Computing Partnership of Japan [23]. The database contains 100 new, original pop songs, 15 royalty-free songs, 50 pieces of classical music, 50 jazz songs, a genre database with 100 songs split among 10 main genres and 33 subcategories, and a musical instrument database with 50 instruments with three performances for each. Corresponding MIDI and text files with lyrics are available. The RWC Music Database has seen somewhat widespread use, but the small amount of songs for each genre; the fact that many of the songs have Japanese lyrics; and the lack of online distribution have altogether limited its use.

uspop2002: The uspop2002 collection by Ellis, Berenzweig and Whitman [24, 25] was one of the first data collections to be distributed. The raw audio was not distributed due to copyright restrictions, but in the hope that it would not upset copyright holders, mel-frequency cepstral coefficients extracted from the music were. While useful for many applications, this does limit the applicability of the collection.

ISMIR 2004: In 2004, several audio description contests were held in connection with the ISMIR conference [26, 27]. In 2005, these contests had developed into the annual Music Information Retrieval Evaluation eXchange (MIREX) evaluations. As part of the 2004 contests, data sets for genre

classification, melody extraction, tempo induction and rhythm classification were released to participants. With the raw audio available, these data sets have been extensively used in research. We have used the ISMIR genre classification training set for Paper B and C, and the rhythm classification set for Paper E.

artist20: Due to the lack of a publicly available artist identification data set, the artist20 collection was released in 2007 by Ellis [28]. It contains six albums by 20 artists each and has significant overlap with the uspop2002 set. However, unlike the uspop2002 set, the artist20 set is distributed as 32 kbps, 16 kHz mono MP3 files.

covers80: The covers80 database was also released by Ellis in 2007, this time to aid the development of cover song identification algorithms [20, 29]. Similar to the artist20 set, this set is also distributed as 32 kbps, 16 kHz mono MP3 files. We have used this data set for developing our cover song identification algorithm in Paper D and the refinement in [19].

MIDI: Finally, for some of our own experiments in Paper A, B, D and F, we have used synthesized MIDI files, allowing full control of the instrumentation and seamless time scaling and transpositions. We have e.g. used this to ensure that our cover song identification algorithms are not affected by key or tempo changes. Although the MIDI files are publicly available, this database has at the time of writing only been used by ourselves.

1.4 Topics in music information retrieval

Before moving on to timbral and melodic similarity in the next two sections, we will briefly describe some other prominent music information retrieval tasks.

Genre classification

Classification of musical genre has received much attention in the music information retrieval community. This interest is quite natural, since genre, together with artist and album names, is one of the most commonly used means of navigating in music collections [2, 30, 31]. Since the first (to the knowledge of the author) example of automated genre classification of sampled audio in 2001 [32], followed by the release of the ISMIR 2004 genre classification data set, there has been an explosion of genre classification algorithms (e.g. [5, 7, 10, 13, 14, 30, 31, 33–38]).

As noted by many authors, musical genres are defined by a mixture of musical as well as cultural properties [2, 30, 31, 39], and as argued in Section 1.1, the performance of genre classification algorithms is therefore inherently limited if

only properties intrinsic to the music are included. In [33], which is an excellent discussion of fundamental genre classification issues, Aucouturier and Pachet have studied different genre taxonomies and found that in e.g. Amazon's genre taxonomy, categories denote such different properties as period (e.g. "60s pop"), topics (e.g. "love song") and country ("Japanese music"). Furthermore, they report that different genre taxonomies are often incompatible, and even within a single genre taxonomy, labels often overlap.

Although occasionally quite high genre classification accuracies are reported, these are often to be taken with a grain of salt. For instance, in the ISMIR 2004 genre classification contest, genre classfication accuracies for a six category genre classification task as high as 84% were reported. However, as argued in [8, 40], this was mainly due to the inclusion of songs by the same artist in both test and training sets, which unintentionally boosted classification accuracies. Also, the test set used in the MIREX evaluation was quite heterogeneous, since approximately 40% of all songs were classical music. Thus, simply guessing that all songs were classical would in itself result in an accuracy of around 40%. Other data sets are more balanced, but the inherent problem, that genres are a cultural as much as a musical phenomenon, persists. Furthermore, many algorithms that work well for genre classification only use short time features on the order of a few tens of milliseconds (e.g. [8, 10, 12] or Paper C), or medium-scale features on the order of e.g. 1 second [13, 14]. On these short time scales, the amount of musical structure that can be modeled is quite limited, and in reality, these methods more likely measure timbral similarity than genre similarity (see Paper A and F). Consequently, e.g. Aucouturier, who originally proposed the algorithm that won the ISMIR 2004 genre classification contest, consistently describes his algorithm as measuring timbral similarity. Indeed, for symbolic genre classification it was shown in [41] that instrumentation is one of the most important features for genre classification.

In our opinion (despite our papers on the topic), genre classification as a research topic in signal processing should be abandoned in favor of specialized tests that directly evaluate the improvements of proposed algorithms. The short time features that only capture timbral similarity, or methods using source separation (e.g. [7, 10]), could e.g. be tested in a polyphonic instrument identification setup that much better shows the capability of algorithms.

Although we are highly sceptical to genre classification as a signal processing task based on sampled audio alone, it might very well be feasible when combining for instance instrument identification algorithms with cultural metadata obtained from internet search engines, wikipedia, online music forums etc. (see e.g. [39, 42–44]). This argument is also put forward by McKay and Fujinaga in [45], where it is argued that automatic genre classification despite much recent criticism indeed is relevant to pursue, for both commercial and scientific reasons. However, one could also argue that applying cultural tags such as genre could

be left for social tagging services combined with audio fingerprinting (see later), and that focus instead should be on developing methods for browsing music based on intrinsic properties such as instrumentation, melody and rhythm, such that music information retrieval systems would provide complementary tools to user-provided descriptions, rather than attempt to imitate them.

Artist recognition

Artist recognition has also received considerable attention (e.g. [28, 42, 46–48]). Artist recognition algorithms are usually designed to identify songs with similar instrumentation, rhythm and/or melody. It is usually a fair assumption that songs by the same artist will be similar in these respects, but it is not guaranteed. Over time, artists that have been active for many years tend to change band members and experiment with new styles. Similar to genre classification, artist recognition is often not a goal in itself, but rather a means to verify that algorithms behave sensibly. When this is the purpose, the albums used for the evaluation are usually chosen such that all songs from one artist are reasonably similar. In 2007, an artist recognition data set became publicly available [28]. Although better defined than genre classification, the artist recognition task is still a mixture of matching similar instrumentation, rhythm and melodies. As such, artist recognition results are interesting when combining e.g. timbre and melody related features as in [28], but such tests should only be considered complementary to experiments that explicitly measure how well the timbral features capture instrumentation and the melodic features capture melody.

Audio tagging/description

Audio tagging is the concept of associating words (tags) to songs, such as done on e.g. Last.fm or MusicBrainz[5]. Automatic audio tagging is a fairly new topic in music information retrieval but has already received much attention (as illustrated by e.g. the special issue [49]). One of the first music-to-text experiments was [50], where the authors attempted to automatically generate song reviews. More recently, Mandel [51] has created an online music tagging game in order to obtain tags for classification experiments. The game is designed as to encourage players to "label songs with original, yet relevant words and phrases that other players agree with"[6]. Among the most popular tags, most either refer to the sound of a song, such as "drums", "guitar" and "male", or to the musical style, such as "rock", "rap", or "soft", while emotional words and words describing the rhythmic content, with the exception of the single word "beat", are almost completely absent [51]. Mandel suggests that the lack of emotional words is caused

[5] `http://last.fm/` and `http://musicbrainz.org/`.
[6] `http://majorminer.org/`

by the difficulty expressing emotions verbally. If this is also the case for the rhythmic descriptors, this suggests that music information retrieval algorithms that helps navigating in music based on rhythmic contents would supplement a word-based music search engine quite well.

Audio fingerprinting

Audio fingerprinting is the task of extracting a hash value, i.e., a fingerprint, that uniquely identifies a recording (for an overview, see the references in [52]). This can e.g. be used to access a database with additional information about the song. MusicBrainz[7] is an example of such a database, where users can enter tags to describe artists, albums or individual songs. With the tendencies of online communities, it does indeed seem possible to tag a significant portion of the world's music, eliminating the need for automated artist and genre classification algorithms before they have even matured. This does not necessarily eliminate the need for timbre, rhythm or melody based search tools, though. On the contrary, it could enable music information retrieval algorithms to only focus on the intrinsic properties of music.

Music similarity

Music similarity, which is the assessment of how similar (or different) two songs are, is often considered the underlying problem that researchers attempt to solve when evaluating their algorithms using genre and artist classification tasks (e.g. [36]) . As pointed out by several researchers [1, 8], music similarity has a number of unfortunate properties, since it is highly subjective and does not obey the triangle inequality. An example given in [8] is a techno version of a classical Mozart concert. Such a song would be similar to both techno songs and classical music, but a user searching for classical music would probably be quite unhappy if served a techno song. As pointed out in [42], music similarity in the general sense is not even a symmetric concept, as it would be more obvious to describe an upcoming band as sounding like e.g. Bruce Springsteen, while you would not describe Bruce Springsteen as sounding like the upcoming band. A simplification of such issues might be to consider e.g. timbre similarity, rhythmic similarity and melodic similarity separately, since even though they do not jointly obey the triangle inequality, each of these aspects may individually do so.

Audio thumbnailing

The concept of thumbnails is well-known from images where a thumbnail refers to a thumbnail-sized preview. Correspondingly, an audio thumbnail summarizes

[7] `http://musicbrainz.org/`

a song in a few seconds of audio, for instance by identifying the chorus or repetitions (see e.g. [53–57]). Despite the applicability of audio thumbnailing when presenting search results to users, this field has only received limited attention.

Rhythmic similarity

Rhythm is a fundamental part of music. Despite this, estimation of rhythmic similarity has not received much attention. As part of the ISMIR audio description contest in 2004 (the precursor to the MIREX evaluations), a rhythm classification contest was held, but it only had one contestant, namely [58]. The few available papers on the subject among others include [59–63], and a review of rhythmic similarity algorithms can be found in e.g. [64] or [65]. Recently, Seyerlehner observed that rhythmic similarity and perceived tempo might be much closer related than previously thought [66], since he was able to match the performance of state of the art tempo induction algorithms with a nearest neighbor classifier using a simple measure of rhythmic distance. As a variation of this, Davies and Plumbley increased tempo estimation accuracies even further by using a simple rhythmic style classifier to obtain a priori probability density functions for different tempi [67].

2 Timbre

Despite the name of this section, it will be almost as much about genre and artist identification as it will be about timbre. The preliminary results of a study by Perrott and Gjerdigen entitled "Scanning the dial: An exploration of factors in the identification of musical style" were presented at the annual meeting of the Society for Music Perception and Cognition in 1999, and it was reported that with music fragments as short as 250 ms, college students in a ten-way forced choice experiment were able to identify the genre of music with an accuracy far better than random. The name of the study, "Scanning the dial", refers to the way listeners scan radio channels to find a channel they like. With such short fragments, almost no melodic or rhythmic information is conveyed, and the study has often been used to argue that only short-time information is needed for genre classification (see [68] for an interesting review of how this preliminary study has influenced the field of music information retrieval). This, and the high success of short-time spectral features in early genre classification works such as [5], has caused much genre classification research to focus on methods that, due to the short time span, in reality only capture timbral information. Consequently, our experiments in Paper A and F show that many features developed for genre classification in reality capture instrumentation.

Returning to timbre, the American Standards Association defines it as "that attribute of auditory sensation in terms of which a listener can judge that two

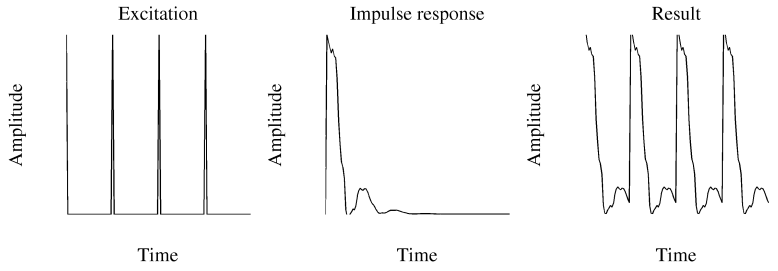


Fig. 5: The excitation impulse train, vocal tract filter and the resulting filtered excitation signal.

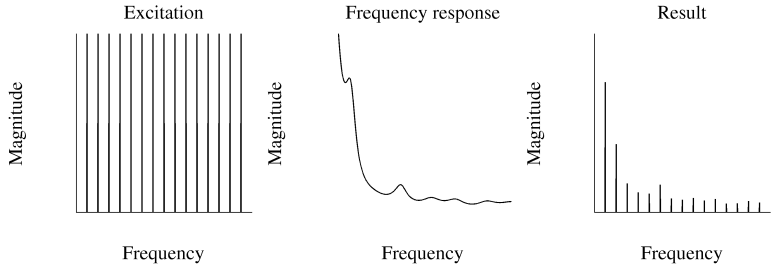


Fig. 6: The excitation impulse train, vocal tract filter and the resulting, filtered excitation signal in the frequency domain.

sounds similarly presented and having the same loudness and pitch are dissimilar" [69]. One can call this an anti-definition, since it defines what timbre is *not*, rather than what it *is*. In the context of music, timbre would e.g. be what distinguishes the sound of the same note played by two different instruments.

In the following, we will present a very simplified view of pitched musical instruments. We start from a model of voiced speech, since it turns out that it is adequate for pitched instruments as well. On a short time scale of a few tens of milliseconds [70], voiced speech is commonly modeled as an excitation signal filtered by the vocal tract impulse response (see Fig. 5 and the frequency domain version in Fig. 6). The fundamental frequency of the excitation signal, an impulse train, determines the pitch of the speech, and the vocal tract impulse response determines the spectral envelope, which among others depend on the vowel being uttered. Pitched instruments can be described by the same short time model. For both speech and musical instruments, the amplitude of the excitation signal slowly changes with time. According to this simplified model, a musical note can be described by the following:

- The fundamental frequency

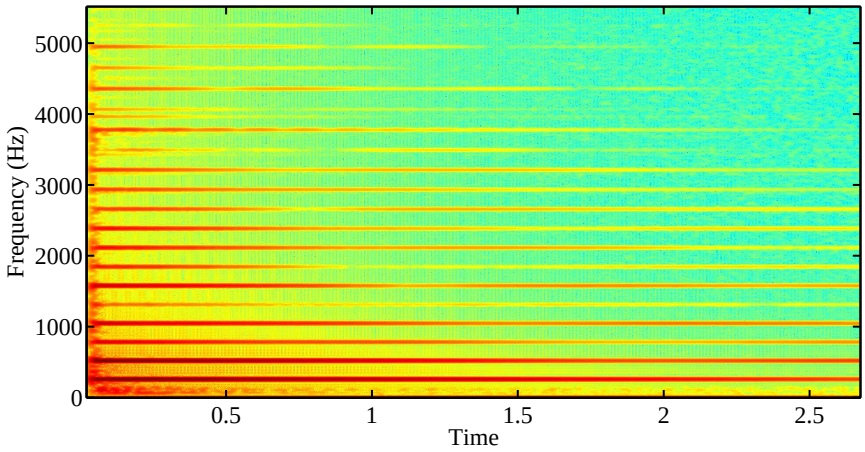


Fig. 7: Spectrogram for a piano C4 note.

- The amplitude of the excitation signal
- The impulse response
- The temporal envelope

The definition of timbre is that it includes everything that characterizes a sound but the volume and pitch, which in the above would be the fundamental frequency and the volume. Thus, according to this simplified model, the timbre of a pitched instrument is characterized by a filter and the temporal envelope. For stationary sounds, the human ear is much more sensitive to the amplitude spectrum of sound than to the phase [71], for which reason only the amplitude (or power) spectrum is frequently considered. We can thus reduce the timbre of a pitched sound to the spectral envelope and the temporal envelope.

This is of course a very simplified model, since other factors also play significant roles. This could be variations in the fundamental frequency (vibrato); the spectral envelope may change over the duration of a note; or the excitation signal could be non-ideal as in e.g. pianos, where the frequencies of the harmonics are slightly higher than integer multiples of the fundamental frequency [71]. Furthermore, we have not even considered the stochastic part of the signal, which also plays a large role (in [72], the stochastic element of notes is actually used to distinguish guitars from pianos). In Fig. 7 and 8, the spectrogram and the power spectrum at different time instants of a piano note are shown, and we see that the harmonics are not exact multiples of the fundamental frequency, and that the spectral envelope changes over time.

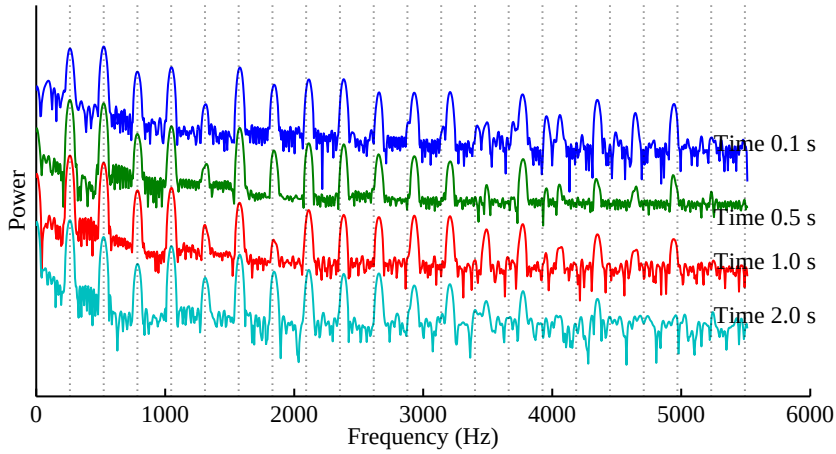


Fig. 8: Power spectrum for a piano C4 note 0.1, 0.5, 1.0 and 1,5 s after onset, respectively. Integer multiples of the fundamental frequency are marked by dashed lines, showing the inharmonicity of a piano. Comparing the top and bottom graphs, we see that high frequency harmonics fade before the lower frequency ones.

2.1 Mel-frequency Cepstral Coefficients

In music information retrieval, only the spectral envelope is commonly considered, and when the temporal envelope is included, it is often in a highly simplified way. By far the most common spectral feature in music information retrieval is the mel-frequency cepstral coefficients (MFCCs) (e.g. [1, 5, 8, 12–14, 30, 31, 37, 38, 73–80]). For other spectral features, see e.g. [81]. MFCCs are short time descriptors of the spectral envelope and are typically computed for audio segments of 10–100 ms [21] as follows:

1. Apply a window function (e.g. the Hamming window) and compute the discrete Fourier transform.
2. Group the frequency bins into M bins equally spaced on the mel frequency scale with 50% overlap.
3. Take the logarithm of the sum of each bin.
4. Compute the discrete cosine transform of the logarithms.
5. Discard high-frequency coefficients from the cosine transform.

In Fig. 9, the spectrum of the piano note in Fig. 8 has been reconstructed from MFCCs. In Paper A, we describe the MFCCs in much more detail.

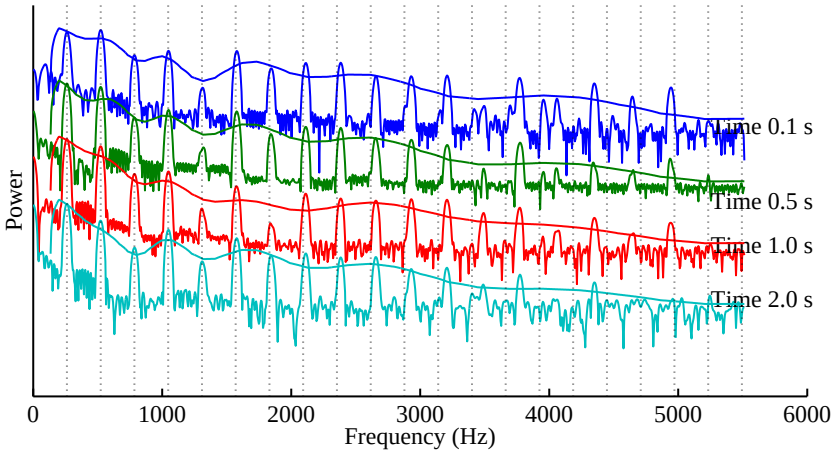


Fig. 9: The spectrum of the piano note in Fig. 8 reconstructed from MFCCs. The reconstructed spectrum is much smoother than the original, thus suppressing the fundamental frequency. Since smoothing is performed on the mel-scale, the spectrum is smoother at high frequencies than at low frequencies.

For trained music information retrieval systems utilizing MFCCs, support vector machines (e.g. [9, 12]) and Gaussian mixture models (e.g. [10, 28]) are quite popular. For untrained systems, it is common to model the MFCCs in a song by either a single, multivariate Gaussian with full covariance matrix or a Gaussian mixture model with diagonal covariance matrices, and then use e.g. the symmetrized Kullback-Leibler divergence between the Gaussian models as a measure of the distance between songs (see Paper A for more details). In Paper B, we have proposed an alternative distance measure that also obeys the triangle inequality. Common to all these approaches is that the resulting model is independent of the temporal order of the MFCCs, and as noted in [17], the model looks the same whether a song is played forwards or backwards. Inspired by the expression "bag of words" from the text retrieval community, this has given such frame-based approaches that ignore temporal information the nickname "bag of frames" approaches. In [82], Aucouturier has exposed listeners to spliced audio, i.e. signals with randomly reordered frames. He finds that it significantly degrades human classification performance for music and concludes that bag of frames approaches have reached the limit of their performance, and that further improvement must be obtained by e.g. incorporating dynamics. He also uses this to conclude that simple variations of the bag of frames approach, such as sophisticated perceptual models, are futile since they cannot compensate for the information loss caused by the splicing.

2.2 Incorporating dynamics

There has been several attempts at incorporating the temporal envelope, i.e., dynamics, in measures of timbre similarity, but this has generally been with limited success for polyphonic music. Aucouturier has performed extensive experiments modeling dynamics by including MFCC delta and acceleration vectors; by using means and variances of MFCCs over 1 s windows similar to [5]; and by using hidden Markov models instead of Gaussian mixture models [17, 21, 83]. He found that although incorporating dynamics increased recognition performance on a monophonic database, it actually decreased performance on a polyphonic database. Flexer [6] also observed that using a hidden Markov model does not increase classification performance despite significantly increasing the log-likelihood of the model. Meng and Ahrendt have experimented with a multivariate autoregressive model for frames of 1.3 s [9], and Scaringella used support vector machines with stacked, delayed inputs [84]. Despite promising results, neither of the two approaches performed significantly better than the static algorithms in the MIREX 2005 genre classification task [14, 85]. For monophonic instrument identification, extensive evaluations can be found in [86].

2.3 Polyphonic timbre similarity

As already argued in Section 1.4, many genre classification systems in reality capture polyphonic timbral similarity rather than genre. In Paper A, we demonstrate this for one particular system. We also conclude that it is much more successful when only a single instrument is playing rather than a mixture of different instruments. This is not surprising, since polyphonic instrument identification is a much harder problem than single-pitch instrument identification. One complication is that while polyphonic music usually is a linear combination of the individual instruments[8], many feature extractors destroy this linearity [87]. A possible solution is to employ a source separation front-end. Although such techniques have not yet been able to perfectly separate polyphonic music into individual instruments, Holzapfel and Stylianou did observe improved genre classification performance by using non-negative matrix factorization [10]. Additionally, several sparse source separation algorithms have been proposed for music. Some of these require previous training (e.g. [87]), some require partial knowledge (e.g. [88]), and yet others are completely data-driven [89]. Not all systems for polyphonic music attempt to separate sources, though. For instance, the system in [90] attempts to recognize ensembles directly in a hierarchical classification scheme.

[8] Non-linear post production effects such as dynamic range compression might actually ruin the additivity.

2.4 The album effect

A problem faced by many timbre-based music information retrieval systems is the so-called album effect [12, 46, 48]. The album effect has its name from artist identification, where it has been observed that if one uses a training set and a test set to estimate the performance of an algorithm, performance is significantly higher if songs from the same album are present among both the test and training data. The same effect has also been observed in genre classification, where songs from the same artist in both the test and the training set significantly increase the observed performance [8, 37, 40, 80], in instrument recognition [18], and we also believe to have re-created the effect in Paper A with synthesized MIDI files, where recognition performance is much higher when test and training data are synthesized using the same sound font.

It is expected that the album effect is caused by post-production effects such as equalization and dynamic range compression [48], but since human listeners hardly seem to notice such post processing, it is unsatisfying that algorithms are so sensitive to this. This discrepancy may be explained by the fact that most algorithms only consider the spectral envelope, whereas e.g. the temporal envelope is also very important to human timbre perception. If one removes the attack part from instrument sounds, it becomes difficult in many cases for humans to tell them apart [71]. Human perception is also quite tolerant to stationary filtering effects. When turning the bass knob on the stereo, we hear that the sound changes, but unless we go to extremes or explicitly pay attention to it, we hardly notice whether the knob is in neutral position. On the other hand, the slow decay of a guitar is quite different from the sudden stop in a banjo. Simple post-production effects such as changing the bass and trepple will change the spectral envelope, but they will not significantly change the temporal envelope. Thus, incorporating dynamics could probably to some extent alleviate the album effect. However, as already stated, it is non-trivial to incorporate dynamics with polyphonic music.

3 Melody

An algorithmic estimate of the perceived similarity between songs' melodies would be very useful for navigating and organizing music collections. However, it is not at all trivial to obtain such as an estimate, and according to the discussion in Section 1.1, such a measure, if it even exists, might be quite subjective. Even if it is subjective, we could hope that it is based on objective properties like tempo or dynamics, such that e.g. latent semantic analysis techniques can be used. Most work on melodic similarity has been on symbolic data (see e.g. the MIREX symbolic melodic similarity task in 2005–2007 or the thesis by Typke [91]) or on the slightly different task of query by humming, where only monophonic

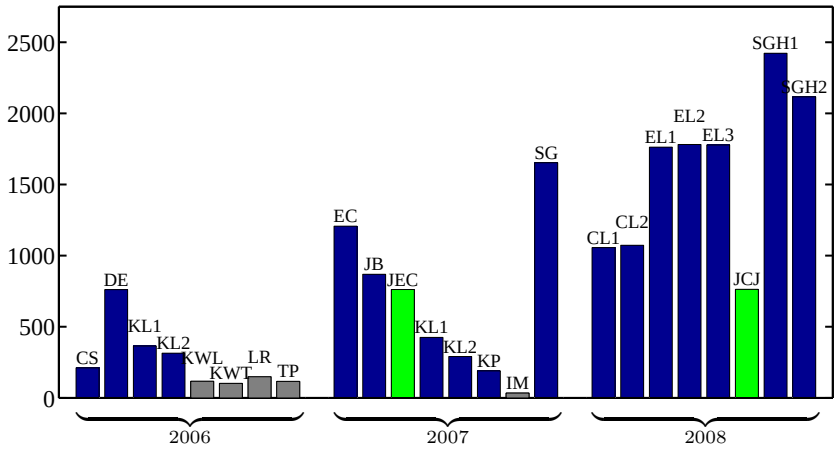


Fig. 10: The number of cover songs identified of the maximum possible 3300 at the MIREX cover song identification task in the years 2006, 2007 and 2008. Gray bars indicate submissions that were developed for more general audio similarity evaluation and not specifically for cover song identification, while the green bars indicate the author's submissions. Analysis of the results for 2006 and 2007 can be found in [92].

melodies are used. Melodic similarity measures for polyphonic, sampled audio have mostly been part of more general music similarity algorithms and, due to the lack of publicly available, annotated melody similarity data sets, have been evaluated in terms of e.g. genre classification (e.g. [5]) and not on data sets designed for testing melodic similarity.

As opposed to general melodic similarity, the related task of audio cover song identification has seen tremendous progress recently. Cover song identification can be considered "melodic fingerprinting" as opposed to estimation of the perceptual similarity, and as such, it is a much simpler task. In 2006, the MIREX Audio Cover Song Identification task was introduced, and as seen in Fig. 10, the identification accuracies have increased significantly each year ever since.

Most cover song retrieval algorithms loosely fit in the framework of Fig. 11, which is a variation of the untrained system block diagram in Fig. 3. With the exception of the contribution of CS in 2006 [93], and with the reservation that details about the algorithm by CL [94] is unavailable, all algorithms specifically designed for cover song retrieval are based on chroma features, which we will describe shortly. Since chroma features are not very compact, all algorithms also include a feature aggregation step, and before comparing the aggregated chroma features from different songs, they have to be normalized with respect to key and tempo. In the following, we will review how different cover song identification systems approach these steps. Since the MIREX 2008 cover song submission by

Table 2: Participants in the MIREX cover song identification tasks.

Year	ID	Participants
2006	CS	Christian Sailer and Karin Dressler
	DE	Daniel P. W. Ellis
	KL	Kyogu Lee
	KWL	Kris West (Likely)
	KWT	Kris West (Trans)
	LR	Thomas Lidy and Andreas Rauber
	TP	Tim Pohle
2007	EC	Daniel P. W. Ellis and Courtenay V. Cotton
	JB	Juan Bello
	JEC	J. H. Jensen, D. P. W. Ellis, M. G. Christensen and S. H. Jensen
	KL	Kyogu Lee
	KP	Youngmoo E. Kim and Daniel Perelstein
	IM	IMIRSEL M2K
	SG	Joan Serrà and Emilia Gómez
2008	CL	C. Cao and M. Li
	EL	A. Egorov and G. Linetsky
	JCJ	J. H. Jensen, M. G. Christensen and S. H. Jensen
	SGH	J. Serrà, E. Gómez and P. Herrera

Egorov and Linetsky use the same strategies as the 2007 submission by Serrà and Gomez [95], we will not discuss the former any further.

3.1 Chromagram

Chromagram features are the melodic equivalent of the timbral MFCC features, and they have been introduced under the name pitch class profiles [96] as well as under the name chromagram [55]. The chromagram is a 12 bin frequency spectrum with one bin for each note on the chromatic scale. Each bin contains the sum of all bins in the full spectrum that are closest to the given note irrespective of octave (see 12). As for MFCCs, details differ among implementations. For instance, to compensate for the Fourier transform's low frequency resolution relative to note bandwidths at low frequencies, the implementation by Ellis is based on the instantaneous frequency [11, 97, 98], while the implementation used by Serrà et al. only use local maxima in the spectrum and also include the first few harmonics of each note [99–101]. In their 2007 submission, they use 36 chroma bins per octave instead of 12, thus dividing the spectrum into 1/3 semitones.

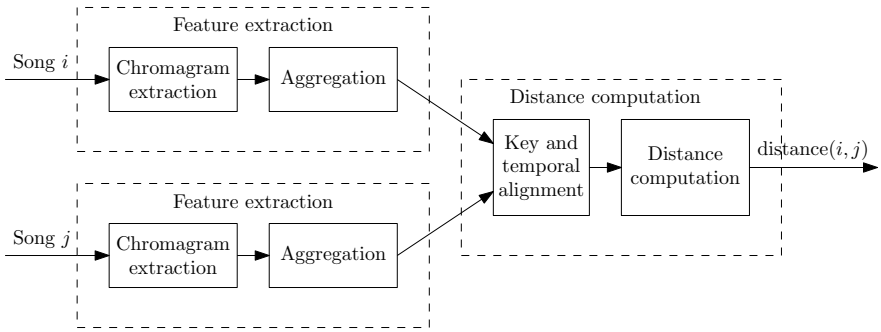


Fig. 11: Block diagram of cover song retrieval algorithms.

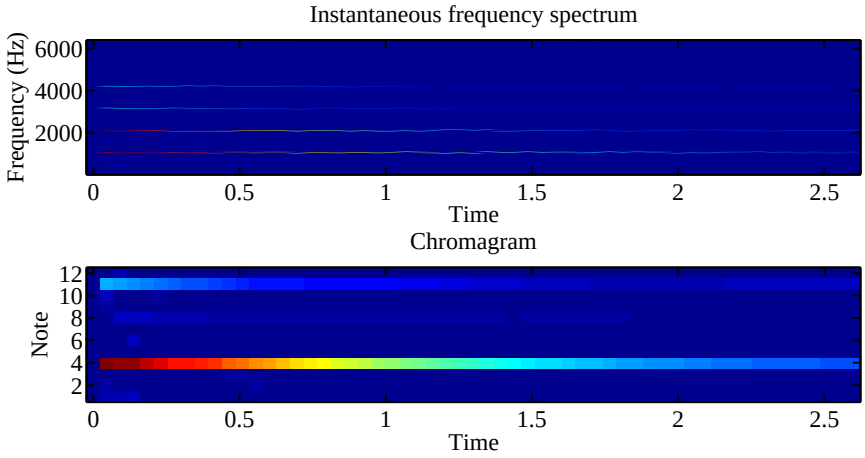


Fig. 12: The instantaneous frequency spectrum and the correpsonding chromagram of the piano note from Fig. 7 and 8.

3.2 Feature aggregation

Several different feature aggregation strategies have been used, some simpler than others. Ellis uses a dynamic beat-tracking algorithm [102] in order to average the chroma vectors over each beat [11], while both Lee and Bello use chord sequences estimated from the chroma features [103, 104]. Serrà et al. simply average the chroma features over approximately 0.5 s [22]. Our submissions, which have some similarity to the approach in [105], summarize the chromagram in a compact feature that is designed to be invariant to changes in tempo (details in Paper D). The aggregated feature of our 2008 submission is also invariant to key changes [19]. Finally, Kim and Perelstein use the state sequence of a hidden Markov model [106].

3.3 Key Alignment

Aligning the keys of two songs that are to be compared is not an insurmountable problem. For instance, the key of both songs can be estimated, whereby the chroma features can be shifted to the same key, as is done by Lee [103]. Serrà et al. has also experimented with this approach [22], but found that it was too sensitive to key estimation errors. Ellis, Bello and our 2007 submission simply compute the distance between songs for all 12 possible transpositions and only keep the shortest distance [11, 104, 107]. Both Kim & Perelstein and our 2008 submission use the autocorrelation function to obtain a representation without any key information, since the autocorrelation function is invariant to offsets [19, 106].

Finally, in Serrà and Gomez's winning 2007 submission, they use use a global chroma vector, which is the average of all the chroma vectors of a song. When aligning two songs, they see how much the global chroma vector of one song has to be shifted to maximize the inner product with the other song's global chroma vector [22]. This optimal shift is then used to align the two songs' chroma vectors. In their 2008 submission, they compute distances for the two best shifts [101] and only keep the shortest distance.

3.4 Temporal Alignment and Distances

In contrast to key alignment, temporal alignment is much more difficult, and it is usually intimately linked to how distances between features are computed. As the measure of similarity between songs, Ellis uses the cross-correlation. To overcome alignment problems, he averages chroma features over each beat, such that the cross-correlation is between songs' beat-synchronous chroma vectors. In the 2007 submission by Ellis and Cotton, to alleviate beat doubling and halving, each song is represented by two different beat estimates, and when comparing two songs, only the maximum correlation obtained among the four combinations is

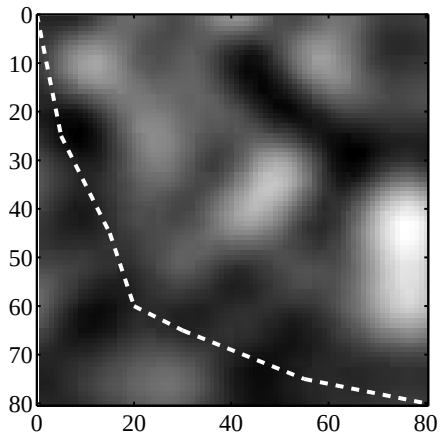


Fig. 13: Conceptual plot of dynamic time warping algorithm. Each pixel (i, j) indicates the distance between song 1 at time i and song 2 at time j with dark colors denoting short distances. The dynamic time warping algorithm finds the path from corner to corner (the dashed line) that minimizes the cumulative distance summed over the path.

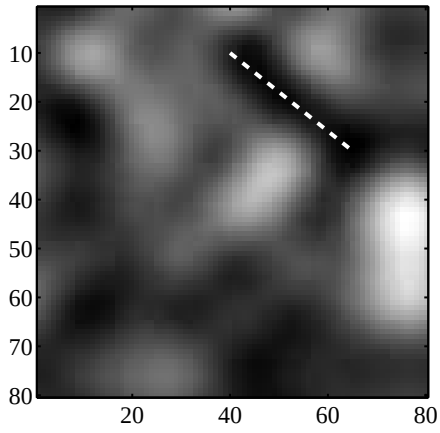


Fig. 14: Unlike the dynamic time warping algorithm in Fig. 13, the dynamic programming local alignment algorithm is not restricted to find a path from corner to corner. It thus finds the best local match (the dashed line) irrespective of the preceding and following distances.

kept. Based on the results of [108], Bello uses string alignment techniques [104]. Interestingly, Bello finds that for string alignment, frame-based chroma features outperforms beat-synchronous chroma features. Lee simply uses dynamic time warping to compare chord sequences [103], while Kim and Perelstein compare histograms of the most likely sequences of states [106]. It is the hope that the hidden markov model state sequences of cover songs will be similar, even though durations of the states will be different. Our 2007 and 2008 submissions both sum statistics of the chromagrams in such a way that differences in tempo are mostly suppressed, which reduces the distance computation to a sum of squared differences. Serrà et al. have performed extensive experiments with different alignment techniques and found that dynamic time warping is suboptimal for cover song identification since it cannot take e.g. the removal of a verse into account [22]. Instead, they found a dynamic programming local alignment algorithm to be superior (see Fig. 13 and 14). The use of this local alignment technique is the most likely reason that the systems by Serrà et al. performed so much better than the competitors.

4 Contribution

The papers that form the main body of this thesis fall into two categories. Paper A to F are concerned with complete music information retrieval frameworks, while on the other hand, Paper G and H treat the low-level problem of fundamental frequency estimation, which is an important part of both transcription and many source separation algorithms. While the papers in the former category use complete music information retrieval frameworks for evaluations, the fundamental frequency estimators in the latter category are evaluated in highly simplified setups with sinusoids in additive Gaussian noise.

Paper A: In this paper, we analyze the modeling of mel-frequency cepstral coefficients by a multivariate Gaussian. By synthesizing MIDI files that suit our needs, we explicitly measure how such a system is affected by different instrument realizations and by music in different keys and with different bitrates and sample rates.

Paper B: Here we also consider systems that model mel-frequency cepstral coefficients by a Gaussian model and introduce an alternative to the Kullback-Leibler divergence that *does* obey the triangle inequality.

Paper C: In this paper, which is the first paper we published on the topic of music information retrieval, we attempted to improve genre classification performance by estimating mel-frequency cepstral coefficients using minimum-variance distortionless response filters (also known as Capon filters). While theoretically this should have resulted in more generalizable

feature vectors, we did not observe any performance increase. This later led us to realize that even if we had observed better performance, it would be farfetched to attribute this to more generalizable features due to the vague assumption of songs from the same genre being similar in only implicitly defined ways. These considerations led to Paper A and F, where we synthesize MIDI files that are tailored to our specific needs.

Paper D: Here we present a cover song identification system based on the simple idea that a change of tempo, which corresponds to stretching the linear time scale by a constant factor, corresponds to a translation by a constant offset on a logarithmic scale.

Paper E: The fifth paper is based on the same idea as the cover song identification algorithm, i.e., that a change of tempo corresponds to translation by an offset. We use this to obtain a representation of rhythmic patterns that is insensitive to minor tempo changes and that has explicit behavior for larger changes.

Paper F: In this paper we use synthesized MIDI songs to experimentally measure the degree to which some common features for genre classification capture instrumentation and melody.

Paper G: Here we derive the joint maximum likelihood estimator of the amplitudes and the noise covariance matrix for a set of sinusoids with known frequencies in colored Gaussian noise. The result is an iterative estimator that surprisingly has the Capon amplitude estimator as a special case.

Paper H: Finally, we compare two variations of the Capon spectral estimator that are modified for fundamental frequency estimation.

At the moment the topics genre classification, artist identification and instrument identification are very intertwined, which makes it difficult to determine if e.g. genre classification improvements are due to improved modeling of instrumentation or melodies, or perhaps due to improved classification algorithms. Our evaluations using MIDI synthesis are small steps towards identifying why algorithms perform as they do, but the community could really need a high-quality freely available polyphonic instrument identification data set.

In the future of music information retrieval, we expect that the line between the intrinsic properties of music and the cultural background information will be drawn sharper. Audio fingerprinting and social tagging services have the potential to deliver much of the cultural information that cannot be extracted directly from the music, and additional cultural information can be retrieved from e.g. internet search engines or from some of the many sites dedicated to music. This reduces and perhaps even eliminates the need to extract e.g. genres

from songs. This will probably increase focus on signal processing algorithms that can extract those intrinsic properties of songs that are tedious to manually label or that are difficult to verbalize. Genre classification will most likely never become the music information retrieval killer application, but search engines based on melodic, rhythmic or timbral similarity do have the potential.

References

[1] A. Berenzweig, "Anchors and hubs in audio-based music similarity," Ph.D. dissertation, Columbia University, New York, Dec. 2007.

[2] F. Pachet and D. Cazaly, "A taxonomy of musical genres," in *In Proc. Content-Based Multimedia Information Access (RIAO) Conf.*, vol. 2, 2000, pp. 1238–1246.

[3] B. Flyvbjerg, *Rationalitet og magt, bind I, Det konkretes videnskab (Rationality and Power, vol. I, Science of the Concrete)*. Copenhagen: Academic Press, 1991.

[4] ——, *Making Social Science Matter: Why Social Inquiry Fails and How It Can Succeed Again*. Cambridge University Press, 2001.

[5] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Processing*, vol. 10, pp. 293–301, 2002.

[6] A. Flexer, E. Pampalk, and G. Widmer, "Hidden markov models for spectral similarity of songs," in *Proc. Int. Conf. Digital Audio Effects*, 2005.

[7] A. S. Lampropoulos, P. S. Lampropoulou, and G. A. Tsihrintzis, "Musical genre classification enhanced by improved source separation technique," in *Proc. Int. Symp. on Music Information Retrieval*, 2005.

[8] E. Pampalk, "Computational models of music similarity and their application to music information retrieval," Ph.D. dissertation, Vienna University of Technology, Austria, Mar. 2006.

[9] A. Meng, P. Ahrendt, J. Larsen, and L. Hansen, "Temporal feature integration for music genre classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1654–1664, July 2007.

[10] A. Holzapfel and Y. Stylianou, "Musical genre classification using nonnegative matrix factorization-based features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 424–434, Feb. 2008.

[11] D. P. W. Ellis and G. Poliner, "Identifying cover songs with chroma features and dynamic programming beat tracking," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2007, pp. 1429–1432.

[12] M. I. Mandel and D. P. W. Ellis, "Song-level features and support vector machines for music classification," in *Proc. Int. Symp. on Music Information Retrieval*, 2005, pp. 594–599.

[13] A. Meng, P. Ahrendt, and J. Larsen, "Improving music genre classification by short-time feature integration," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Mar. 2005, pp. 497–500.

[14] P. Ahrendt and A. Meng, "Music genre classification using the multivariate AR feature integration model," in *Music Information Retrieval Evaluation eXchange*, 2005.

[15] M. I. Mandel, G. Poliner, and D. P. W. Ellis, "Support vector machine active learning for music retrieval," *ACM Multimedia Systems Journal*, vol. 12, no. 1, pp. 3 – 13, April 2006.

[16] Wikipedia. (2009, Feb.) K-nearest neighbor algorithm. [Online]. Available: http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

[17] J.-J. Aucouturier, "Ten experiments on the modelling of polyphonic timbre," Ph.D. dissertation, University of Paris 6, France, Jun. 2006.

[18] A. Livshin and X. Rodet, "The importance of cross database evaluation in sound classification," in *Proc. Int. Symp. on Music Information Retrieval*, 2003.

[19] J. H. Jensen, M. G. Christensen, and S. H. Jensen, "A chroma-based tempo-insensitive distance measure for cover song identification using the 2D autocorrelation function," in *Music Information Retrieval Evaluation eXchange*, 2008.

[20] D. P. W. Ellis. (2007) The "covers80" cover song data set. [Online]. Available: http://labrosa.ee.columbia.edu/projects/coversongs/covers80/

[21] J.-J. Aucouturier and F. Pachet, "Improving timbre similarity: How high's the sky?" *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.

[22] J. Serra, E. Gomez, P. Herrera, and X. Serra, "Chroma binary similarity and local alignment applied to cover song identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 6, pp. 1138–1151, Aug. 2008.

[23] M. Goto, "Development of the RWC music database," in *Proc. Int. Congress on Acoustics*, 2004, pp. I–553–556.

[24] A. Berenzweig, B. Logan, D. P. W. Ellis, and B. Whitman, "A large-scale evaluation of acoustic and subjective music similarity measures," in *Proc. Int. Symp. on Music Information Retrieval*, 2003.

[25] D. P. W. Ellis, A. Berenzweig, and B. Whitman. (2003) The "uspop2002" pop music data set. [Online]. Available: http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html

[26] P. Cano, E. Gómez, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S. Streich, and N. Wack, "Ismir 2004 audio description contest," Music Technology Group of the Universitat Pompeu Fabra, Tech. Rep., 2006. [Online]. Available: http://www.iua.upf.edu/mtg/files/publications/MTG-TR-2006-02.pdf

[27] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE Trans. Audio, Speech, Language Processing*, vol. 14, no. 5, 2006.

[28] D. P. W. Ellis, "Classifying music audio with timbral and chroma features," in *Proc. Int. Symp. on Music Information Retrieval*, Vienna, Austria, 2007.

[29] D. P. W. Ellis and C. V. Cotton, "The 2007 LabROSA cover song detection system," in *Music Information Retrieval Evaluation eXchange*, 2007.

[30] P. Ahrendt, "Music genre classification systems," Ph.D. dissertation, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2006.

[31] A. Meng, "Temporal feature integration for music organisation," Ph.D. dissertation, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2006.

[32] G. Tzanetakis, G. Essl, and P. Cook, "Automatic musical genre classification of audio signals," in *Proc. Int. Symp. on Music Information Retrieval*, 2001, pp. 205–210.

[33] J.-J. Aucouturier and F. Pachet, "Representing musical genre: A state of the art," *J. New Music Research*, vol. 32, no. 1, 2003.

[34] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer, "Evaluating rhythmic descriptors for musical genre classification," in *Proc. Int. AES Conference*, London, UK, 2004, p. 196ŋ204.

[35] A. Meng and J. Shawe-Taylor, "An investigation of feature models for music genre classification using the support vector classifier," in *Proc. Int. Symp. on Music Information Retrieval*, Sep. 2005.

[36] E. Pampalk, A. Flexer, and G. Widmer, "Improvements of audio-based music similarity and genre classification," in *Proc. Int. Symp. on Music Information Retrieval*, London, UK, Sep. 2005.

[37] A. Flexer, "Statistical evaluation of music information retrieval experiments," Institute of Medical Cybernetics and Artificial Intelligence, Medical University of Vienna, Tech. Rep., 2005.

[38] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl, "Aggregate features and AdaBoost for music classification," *Machine Learning*, vol. 65, no. 2–3, pp. 473–484, Dec. 2006.

[39] B. Whitman and P. Smaragdis, "Combining musical and cultural features for intelligent style detection," in *Proc. Int. Symp. on Music Information Retrieval*, 2002, pp. 47–52.

[40] A. Flexer, "A closer look on artist filters for musical genre classification," in *Proc. Int. Symp. on Music Information Retrieval*, 2007, pp. 341–344.

[41] C. McKay and I. Fujinaga, "Automatic music classification and the importance of instrument identification," in *Proc. of the Conf. on Interdisciplinary Musicology*, 2005.

[42] D. P. W. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence, "The quest for ground truth in musical artist similarity," in *Proc. Int. Symp. on Music Information Retrieval*, 2002, pp. 170–177.

[43] M. Schedl, P. Knees, and G. Widmer, "Discovering and visualizing prototypical artists by web-based co-occurrence analysis," in *Proc. Int. Symp. on Music Information Retrieval*, Sep. 2005.

[44] P. Knees, M. Schedl, and T. Pohle, "A deeper look into web-based classification of music artists," in *Proc. Workshop on Learning the Semantics of Audio Signals*, Paris, France, June 2008.

[45] C. McKay and I. Fujinaga, "Musical genre classification: Is it worth pursuing and how can it be improved?" in *Proc. Int. Symp. on Music Information Retrieval*, 2006.

[46] B. Whitman, G. Flake, and S. Lawrence, "Artist detection in music with minnowmatch," in *Proc. IEEE Workshop on Neural Networks for Signal Processing*, 2001, pp. 559–568.

[47] A. Berenzweig, D. P. W. Ellis, and S. Lawrence, "Anchor space for classification and similarity measurement of music," in *Proc. International Conference on Multimedia and Expo ICME '03*, vol. 1, 2003, pp. I–29–32 vol.1.

[48] Y. E. Kim, D. S. Williamson, and S. Pilli, "Towards quantifying the "album effect" in artist identification," in *Proc. Int. Symp. on Music Information Retrieval*, 2006.

[49] "From genres to tags: Music information retrieval in the era of folksonomies," *J. New Music Research*, vol. 37, no. 2, 2008.

[50] B. Whitman and D. P. W. Ellis, "Automatic record reviews," in *Proc. Int. Symp. on Music Information Retrieval*, 2004.

[51] M. I. Mandel and D. P. W. Ellis, "A web-based game for collecting music metadata," in *Proc. Int. Symp. on Music Information Retrieval*, 2007.

[52] P. Cano, "Content-based audio search: From fingerprinting to semantic audio retrieval," Ph.D. dissertation, University Pompeu Fabra, Barcelona, Spain, 2006.

[53] B. Logan and S. Chu, "Music summarization using key phrases," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP '00*, vol. 2, 2000, pp. II749–II752 vol.2.

[54] G. Tzanetakis, "Manipulation, analysis and retrieval systems for audio signals," Ph.D. dissertation, Princeton University, Jun. 2002.

[55] M. A. Bartsch and G. H. Wakefield, "To catch a chorus: using chroma-based representations for audio thumbnailing," in *Proc. IEEE Workshop on Appl. of Signal Process. to Aud. and Acoust.*, 2001, pp. 15 – 18.

[56] M. Bartsch and G. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 96–104, 2005.

[57] M. Goto, "A chorus section detection method for musical audio signals and its application to a music listening station," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1783–1794, Sept. 2006.

[58] T. Lidy, "Evaluation of new audio features and their utilization in novel music retrieval applications," Master's thesis, Vienna University of Technology, Dec. 2006.

[59] J. Foote, M. Cooper, and U. Nam, "Audio retrieval by rhythmic similarity," in *Proc. Int. Symp. on Music Information Retrieval*, 2002, pp. 265–266.

[60] J. Paulus and A. Klapuri, "Measuring the similarity of rhythmic patterns," in *Proc. Int. Symp. on Music Information Retrieval*, 2002, pp. 150–156.

[61] S. Dixon, F. Gouyon, and G. Widmer, "Towards characterisation of music via rhythmic patterns," in *Proc. Int. Symp. on Music Information Retrieval*, 2004, pp. 509–516. [Online]. Available: http://ismir2004.ismir.net/proceedings/p093-page-509-paper165.pdf

[62] G. Peeters, "Rhythm classification using spectral rhythm patterns," in *Proc. Int. Symp. on Music Information Retrieval*, 2005, pp. 644–647.

[63] N. Scaringella, "Timbre and rhythmic trap-tandem features for music information retrieval," in *Proc. Int. Symp. on Music Information Retrieval*, 2008, pp. 626–631.

[64] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer, "A review of automatic rhythm description systems," *Computer Music Journal*, vol. 29, no. 1, pp. 34–54, 2005.

[65] F. Gouyon, "A computational approach to rhythm description — Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing," Ph.D. dissertation, University Pompeu Fabra, Barcelona, Spain, November 2005.

[66] K. Seyerlehner, G. Widmer, and D. Schnitzer, "From rhythm patterns to perceived tempo," in *Proc. Int. Symp. on Music Information Retrieval*, Vienna, Austria, 2008, pp. 519–524.

[67] M. E. P. Davies and M. D. Plumbley, "Exploring the effect of rhythmic style classification on automatic tempo estimation," in *Proc. European Signal Processing Conf.*, 2008.

[68] J.-J. Aucouturier and E. Pampalk, "From genres to tags: A little epistemology of music information retrieval research," *J. New Music Research*, vol. 37, no. 2, 2008.

[69] *Acoustical Terminology SI*, New York: American Standards Association Std., Rev. 1-1960, 1960.

[70] J. John R. Deller, J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*, 2nd ed. Wiley-IEEE Press, 1999.

[71] T. D. Rossing, F. R. Moore, and P. A. Wheeler, *The Science of Sound*, 3rd ed. Addison-Wesley, 2002.

[72] D. Fragoulis, C. Papaodysseus, M. Exarhos, G. Roussopoulos, T. Panagopoulos, and D. Kamarotos, "Automated classification of piano-guitar notes," *IEEE Trans. Audio, Speech, Language Processing*, vol. 14, no. 3, pp. 1040– 1050, May 2006.

[73] J. T. Foote, "Content-based retrieval of music and audio," in *Multimedia Storage and Archiving Systems II, Proc. of SPIE*, 1997, pp. 138–147.

[74] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proc. IEEE Int. Conf. Multimedia Expo*, Tokyo, Japan, 2001, pp. 745 – 748.

[75] J.-J. Aucouturier and F. Pachet, "Finding songs that sound the same," in *Proc. of IEEE Benelux Workshop on Model based Processing and Coding of Audio*, University of Leuven, Belgium, November 2002.

[76] E. Pampalk, "A Matlab toolbox to compute music similarity from audio," in *Proc. Int. Symp. on Music Information Retrieval*, 2004, pp. 254–257.

[77] ——, "Speeding up music similarity," in *2nd Annual Music Information Retrieval eXchange*, London, Sep. 2005.

[78] M. Levy and M. Sandler, "Lightweight measures for timbral similarity of musical audio," in *Proc. ACM Multimedia*. New York, NY, USA: ACM, 2006, pp. 27–36.

[79] J.-J. Aucouturier, F. Pachet, and M. Sandler, ""the way it sounds": timbre models for analysis and retrieval of music signals," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1028–1035, 2005.

[80] J. H. Jensen, M. G. Christensen, M. N. Murthi, and S. H. Jensen, "Evaluation of MFCC estimation techniques for music similarity," in *Proc. European Signal Processing Conf.*, Florence, Italy, 2006.

[81] D. Stowell and M. D. Plumbley, "Robustness and independence of voice timbre features under live performance acoustic degradations," in *Proc. Int. Conf. Digital Audio Effects*, 2008.

[82] J.-J. Aucouturier. (2008, Dec.) A day in the life of a Gaussian mixture model. [Online]. Available: http://www.jj-aucouturier.info/papers/ISMIR-2008.pdf

[83] J.-J. Aucouturier and F. Pachet, "The influence of polyphony on the dynamical modelling of musical timbre," *Pattern Recognition Letters*, vol. 28, no. 5, pp. 654–661, 2007.

[84] N. Scaringella and G. Zoia, "On the modeling of time information for automatic genre recognition systems in audio signals," in *Proc. Int. Symp. on Music Information Retrieval*, 2005.

[85] N. Scaringella and D. Mlynek, "A mixture of support vector machines for audio classification," in *Music Information Retrieval Evaluation eXchange*, 2005.

[86] C. Joder, S. Essid, and G. Richard, "Temporal integration for audio classification with application to musical instrument classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 174–186, 2009.

[87] G. Richard, P. Leveau, L. Daudet, S. Essid, and B. David, "Towards polyphonic musical instruments recognition," in *Proc. Int. Congress on Acoustics*, 2007.

[88] H. Laurberg, M. N. Schmidt, M. G. Christensen, and S. H. Jensen, "Structured non-negative matrix factorization with sparsity patterns," in *Rec. Asilomar Conf. Signals, Systems and Computers*, 2008.

[89] S. Abdallah and M. Plumbley, "Unsupervised analysis of polyphonic music by sparse coding," *IEEE Trans. Neural Networks*, vol. 17, no. 1, pp. 179–196, 2006.

[90] S. Essid, G. Richard, and B. David, "Instrument recognition in polyphonic music based on automatic taxonomies," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 68–80, 2006.

[91] R. Typke, "Music retrieval based on melodic similarity," Ph.D. dissertation, Utrecht University, The Netherlands, 2007.

[92] J. S. Downie, M. Bay, A. F. Ehmann, and M. C. Jones, "Audio cover song identification: MIREX 2006-2007 results and analyses," in *Proc. Int. Symp. on Music Information Retrieval*, 2008, pp. 468–473.

[93] C. Sailer and K. Dressler, "Finding cover songs by melodic similarity," in *Music Information Retrieval Evaluation eXchange*, 2006.

[94] C. Cao and M. Li, "Cover version detection for audio music," in *Music Information Retrieval Evaluation eXchange*, 2008.

[95] A. Egorov and G. Linetsky, "Cover song identification with IF-F0 pitch class profiles," in *Music Information Retrieval Evaluation eXchange*, 2008.

[96] T. Fujishima, "Real-time chord recognition of musical sound: A system using common lisp music," in *Proc. Int. Computer Music Conf.*, 1999, pp. 464–467.

[97] D. P. W. Ellis, "Identifying 'cover songs' with beat-synchronous chroma features," in *Music Information Retrieval Evaluation eXchange*, 2006.

[98] F. Charpentier, "Pitch detection using the short-term phase spectrum," in *Proc. IEEE International Conference on ICASSP '86. Acoustics, Speech, and Signal Processing*, vol. 11, Apr 1986, pp. 113–116.

[99] E. G. Gutiérrez, "Tonal description of music audio signals," Ph.D. dissertation, Universitat Pompeu Fabra, Spain, 2006.

[100] J. Serra and E. Gomez, "Audio cover song identification based on tonal sequence alignment," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2008*, March 31 2008–April 4 2008, pp. 61–64.

[101] J. Serrà, E. Gómez, and P. Herrera, "Improving binary similarity and local alignment for cover song detection," in *Music Information Retrieval Evaluation eXchange*, 2008.

[102] D. P. W. Ellis, "Beat tracking with dynamic programming," in *Music Information Retrieval Evaluation eXchange*, 2006.

[103] K. Lee, "Identifying cover songs from audio using harmonic representation," in *Music Information Retrieval Evaluation eXchange*, 2006.

[104] J. P. Bello, "Audio-based cover song retrieval using approximate chord sequences: Testing shifts, gaps, swaps and beats," in *Proc. Int. Symp. on Music Information Retrieval*, 2007, pp. 239–244.

[105] T. Lidy and A. Rauber, "Combined fluctuation features for music genre classification," in *Music Information Retrieval Evaluation eXchange*, 2005.

[106] Y. E. Kim and D. Perelstein, "MIREX 2007: Audio cover song detection using chroma features and a hidden markov model," in *Music Information Retrieval Evaluation eXchange*, 2007.

[107] J. H. Jensen, M. G. Christensen, D. P. W. Ellis, and S. H. Jensen, "A tempo-insensitive distance measure for cover song identification based on chroma features," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Las Vegas, USA, 2008, pp. 2209–2212.

[108] M. Casey and M. Slaney, "The importance of sequences in musical similarity," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2006*, vol. 5, 2006, pp. V–V.

Paper A

Quantitative Analysis of a Common Audio Similarity Measure

Jesper Højvang Jensen, Mads Græsbøll Christensen,
Daniel P.W. Ellis, and Søren Holdt Jensen

This paper has been published in
IEEE Transactions on Audio, Speech, and Language Processing,
vol. 17(4), pp. 693–703, May 2009.

The layout has been revised.

Abstract

For music information retrieval tasks, a nearest neighbor classifier using the Kullback-Leibler divergence between Gaussian mixture models of songs' mel-frequency cepstral coefficients is commonly used to match songs by timbre. In this paper, we analyze this distance measure analytically and experimentally by the use of synthesized MIDI files, and we find that it is highly sensitive to different instrument realizations. Despite the lack of theoretical foundation, it handles the multipitch case quite well when all pitches originate from the same instrument, but it has some weaknesses when different instruments play simultaneously. As a proof of concept, we demonstrate that a source separation frontend can improve performance. Furthermore, we have evaluated the robustness to changes in key, sample rate and bitrate.

1 Introduction

Mel-frequency cepstral coefficients (MFCCs) are extensively used in music information retrieval algorithms [1–12]. Originating in speech processing, the MFCCs were developed to model the spectral envelope while suppressing the fundamental frequency. Together with the temporal envelope, the spectral envelope is one of the most salient components of timbre [13, 14], which is "that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness and pitch are dissimilar" [15], i.e., what makes the same note played with different instruments sound different. Thus, the MFCCs in music information retrieval applications are commonly used to model the timbre. However, even though MFCCs have experimentally been shown to perform well in instrument recognition, artist recognition and genre classification [7, 8, 16], a number of questions remain unanswered. For instance, being developed for speech recognition in a single speaker environment, it is not obvious how the MFCCs are affected by different instruments playing simultaneously and by chords where the fundamental frequencies have near-integer ratios. Furthermore, as shown in [17], MFCCs are sensitive to the spectral perturbations that result from low bitrate audio compression.

In this paper, we address these issues and more. We analyze the behaviour of the MFCCs when either a single instrument or different instruments play several notes simultaneously, thus violating the underlying assumption of a single voice. In relation to the album effect [18], where MFCC-based distance measures in artist recognition rate songs from the same album as much more similar than songs by the same artist from different albums, we investigate how MFCCs are affected by different realizations of the same instrument. Finally, we investigate how MFCCs are affected by transpositions, different sample rates and different

bitrates, since this is relevant in practical applications. A transposed version of a song, e.g. a live version that is played in a different key than the studio version, is usually considered similar to the orignal, and collections of arbitrary music, such as encountered by an internet search engine, will inevitably contain songs with different sample rates and bitrates.

To analyze these topics, we use MIDI synthesis, for reasons of tractability and reproducibility, to fabricate wave signals for our experiments, and we employ the distance measure proposed in [4] that extracts MFCCs and trains a Gaussian mixture model for each song and uses the symmetrized Kullback-Leibler divergence between the models as distance measure. A nearest neighbor classification algorithm using this approach won the International Conference on Music Information Retrieval (ISMIR) genre classification contest in 2004 [6]. Genre classification is often not considered a goal in itself, but rather an indirect means to verify the actual goal, which is a measure of similarity between songs. In most comparisons on tasks such as genre identification, distributions of MFCC features have performed as well or better than all other features considered—a notable result [7, 8]. Details of the system, such as the precise form or number of MFCCs used, or the particular mechanism used to represent and compare MFCC distributions, appear to have only a secondary influence. Thus, the distance measure studied in this paper, a particular instance of a system for comparing music audio based on MFCC distributions, is both highly representative of most current work in music audio comparison, and is likely close to or equal to the state of the art in most tasks of this kind.

In Section 2, we review MFCCs, Gaussian modelling and computation of the symmetrized Kullback-Leibler divergence. In Section 3, we describe the experiments before discussing the results in Section 4 and giving the conclusion in Section 5.

2 A Mel-Frequency Cepstral Coefficients based Timbral Distance Measure

In the following we describe the motivation behind the MFCCs, mention some variations of the basic concept, discuss their applicability to music and discuss the use of the Kullback-Leibler divergence between multivariate Gaussian mixture models as a distance measure between songs.

2.1 Mel-Frequency Cepstral Coefficients

MFCCs were introduced as a compact, perceptually based representation of speech frames [19]. They are computed as follows:

1. Estimate the amplitude or power spectrum of 20–30 ms of speech.

2. Group neighboring frequency bins into overlapping triangular bands with equal bandwidth according to the mel-scale.

3. Sum the contents of each band.

4. Compute the logarithm of each sum.

5. Compute the discrete cosine transform of the bands.

6. Discard high-order coefficients from the cosine transform.

Most of these steps are perceptually motivated, but some steps also have a signal processing interpretation. The signal is divided into 20–30 ms blocks because speech is approximately stationary within this time scale. Grouping into bands and summing mimics the difficulty in resolving two tones closely spaced in frequency, and the logarithm approximates the human perception of loudness. The discrete cosine transform, however, does not directly mimic a phenomenon in the human auditory system, but is instead an approximation to the Karhunen-Loève transform in order to obtain a compact representation with minimal correlation between different coefficients.

As the name of the MFCCs imply, the last three steps can also be interpreted as homomorphic deconvolution in the cepstral domain to obtain the spectral envelope (see, e.g., [20]). Briefly, this approach employs the common model of voice as glottal excitation filtered by a slowly-changing vocal tract, and attempts to separate these two components. The linear filtering becomes multiplication in the Fourier domain, which then turns into addition after the logarithm. The final Fourier transform, accomplished by the discrete cosine transform, retains linearity but further allows separation between the vocal tract spectrum, which is assumed smooth in frequency and thus ends up being represented by the low-index cepstral coefficients, and the harmonic spectrum of the excitation, which varies rapidly with frequency and falls predominantly into higher cepstral bins. These are discarded, leaving a compact feature representation that describes the vocal tract characteristics with little dependence on the fine structure of the excitation (such as its period). For a detailed description of homomorphic signal processing see [21], and for a discussion of the statistical properties of the cepstrum see [22]. For a discussion of using the MFCCs as a model for perceptual timbre space for static sounds, see [23].

2.2 Variations

When computing MFCCs from a signal, there are a number of free parameters. For instance, both the periodogram, linear prediction analysis, the Capon spectral estimator and warped versions of the latter two have been used to estimate the spectrum, and the number of mel-distributed bands and their lower and

upper cut-off frequency may also differ. For speech recognition, comparisons of different such parameters can be found in [24] and [25]. For music, less exhaustive comparisons can be found in [5] and [12]. It is also an open question how many coefficients should be kept after the discrete cosine transform. According to [17], the first five to fifteen are commonly used. In [26], as many as 20 coefficients, excluding the 0th coefficient, are used with success. In the following, we will use the term "MFCC order" to refer to the number of coefficients that are kept. Another open question is whether to include the 0th coefficient. Being the DC value, the 0th coefficient is the average of the logarithm of the summed contents of the triangular bands, and it can thus be interpreted as the loudness averaged over the triangular bands. On the one hand, volume may be useful for modelling a song, while on the other hand it is subject to arbitrary shifts (i.e., varying the overall scale of the waveform) and does not contain information about the spectral shape as such.

2.3 Applicability to Music

In [27], it is verified that the mel-scale is preferable to a linear scale in music modelling, and that the discrete cosine transform does approximate the Karhunen-Loève transform. However, a number of uncertainties remain. In particular, the assumed signal model consisting of one excitation signal and a filter only applies to speech. In polyphonic music there may, unlike in speech, be several excitation signals with different fundamental frequencies and different filters. Not only may this create ambiguity problems when estimating which instruments the music was played by, since it is not possible to uniquely determine how each source signal contributed to the spectral envelopes, but the way the sources combine is also very nonlinear due to the logarithm in step 4. Furthermore, it was shown in [17] that MFCCs are sensitive to the spectral perturbations that are introduced when audio is compressed at low bitrates, mostly due to distortion at higher frequencies. However, it was not shown whether this actually affects instrument or genre classification performance. A very similar issue is the sampling frequency of the music that the MFCCs are computed from. In a real world music collection, all music may not have the same sampling frequency. A downsampled signal would have very low energy in the highest mel-bands, leaving the logarithm in step 4 in the MFCC computation either undefined or at least approaching minus infinity. In practical applications, some minimal (floor) value is imposed on channels containing little or no energy. When the MFCC analysis is applied over a bandwidth greater than that remaining in the compressed waveform, this amounts to imposing a rectangular window on the spectrum, or, equivalently, convolving the MFCCs with a sinc function. We will return to these issues in Section 3.

2.4 Modelling MFCCs by Gaussian Mixture Models

Storing the raw MFCCs would take up a considerable amount of space, so the MFCCs from each song are used to train a parametric, statistical model, namely a multivariate Gaussian mixture model. As distance measure between the Gaussian mixture models, we use the symmetrized Kullback-Leibler divergence. This approach was presented in [4], but both [2] and [28] have previously experimented with very similar approaches. The probability density function for a random variable $\boldsymbol{x}$ modelled by a Gaussian mixture model with K mixtures is given by

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} c_k \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_k|}} \exp\Big(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}{\boldsymbol{\Sigma}_k}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\Big), \tag{A.1}$$

where K is the number of mixtures and $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and c_k are the mean, covariance matrix and weight of the k'th Gaussian, respectively. For $K = 1$, the maximum-likelihood estimates of the mean and covariance matrix are given by [29]

$$\boldsymbol{\mu}_{\mathrm{ML}} = \frac{1}{M}\sum_{n=1}^{M} \boldsymbol{x}_n \tag{A.2}$$

and

$$\boldsymbol{\Sigma}_{\mathrm{ML}} = \frac{1}{M}\sum_{n=1}^{M} (\boldsymbol{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})(\boldsymbol{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})^{\mathrm{T}}. \tag{A.3}$$

For $K > 1$, the k-means algorithm followed by the expectation-maximization algorithm (see [30, 31]) is typically used to train the weights c_k, means $\boldsymbol{\mu}_k$ and covariance matrices $\boldsymbol{\Sigma}_k$. As mentioned, we use the symmetrized Kullback-Leibler divergence between the Gaussian mixtures as distance measure between two songs. The Kullback-Leibler divergence is an asymmetric information theoretic measure of the distance between two probability density functions. The Kullback-Leibler divergence between $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$, $d_{\mathrm{KL}}(p_1, p_2)$, is given by

$$d_{\mathrm{KL}}(p_1, p_2) = \int p_1(\boldsymbol{x}) \log \frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})} d\boldsymbol{x}. \tag{A.4}$$

For discrete random variables, $d_{\mathrm{KL}}(p_1, p_2)$ is the penalty of designing a code that describes data with distribution $p_2(\boldsymbol{x})$ with shortest possible length but instead use it to encode data with distribution $p_1(\boldsymbol{x})$ [32]. If $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$ are close, the penalty will be small and vice versa. For two multivariate Gaussian distributions, $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$, the Kullback-Leibler divergence is given in closed

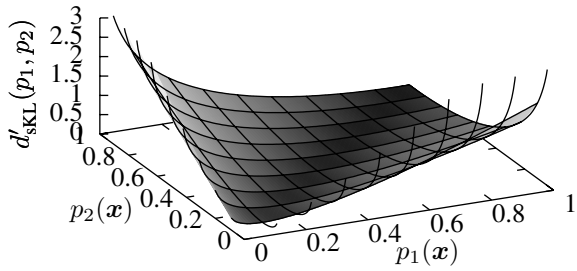


Fig. A.1: Symmetrized Kullback-Leibler divergence. When either $p_1(\boldsymbol{x})$ or $p_1(\boldsymbol{x})$ approach zero, $d'_{\mathrm{sKL}}(p_1, p_2)$ approaches infinity.

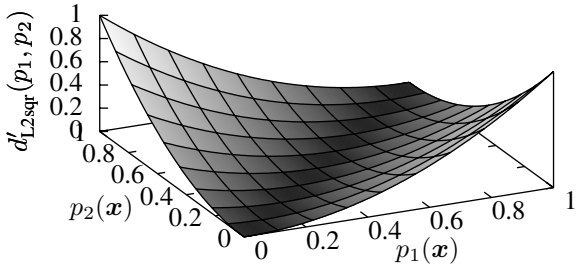


Fig. A.2: The squared L2 distance. Note that unlike $d'_{\mathrm{sKL}}(p_1, p_2)$ in Fig. A.1, $d'_{\mathrm{L2sqr}}(p_1, p_2)$ behaves nicely when $p_1(\boldsymbol{x})$ or $p_2(\boldsymbol{x})$ approach zero.

form by

$$d_{\mathrm{KL}}(p_1, p_2) = \frac{1}{2}\Bigg[\log\left(\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|}\right) + \mathrm{tr}({\boldsymbol{\Sigma}_1}^{-1}\boldsymbol{\Sigma}_2) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^{\mathrm{T}}{\boldsymbol{\Sigma}_1}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) - D\Bigg] \tag{A.5}$$

where D is the dimensionality of $\boldsymbol{x}$. For Gaussian mixtures, a closed form expression for $d_{\mathrm{KL}}(p_1, p_2)$ does not exist, and it must be estimated e.g. by stochastic integration or closed form approximations [10, 33, 34]. To obtain a symmetric distance measure, we use $d_{\mathrm{sKL}}(p_1, p_2) = d_{\mathrm{KL}}(p_1, p_2) + d_{\mathrm{KL}}(p_2, p_1)$.

Collecting the two Kullback-Leibler divergences under a single integral, we can directly see how different values of $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$ affect the resulting distance:

$$d_{\mathrm{sKL}}(p_1, p_2) = \int d'_{\mathrm{sKL}}(p_1(\boldsymbol{x}), p_2(\boldsymbol{x}))d\boldsymbol{x}, \tag{A.6}$$

where

$$d'_{\mathrm{sKL}}(p_1(\boldsymbol{x}), p_2(\boldsymbol{x})) = (p_1(\boldsymbol{x}) - p_2(\boldsymbol{x})) \log \frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})}. \tag{A.7}$$

In Fig. A.1, $d'_{\mathrm{sKL}}(p_1(\boldsymbol{x}), p_2(\boldsymbol{x}))$ is shown as a function of $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$. From the figure and (A.7), it is seen that for $d_{\mathrm{sKL}}(p_1, p_2)$ to be large, there has to be $\boldsymbol{x}$ where both the difference and the ratio between $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$ is large. High values are obtained when only one of $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$ approach zero. In comparison, consider the square of the L2 distance, which is given by

$$d_{\mathrm{L2}}(p_1, p_2)^2 = \int d'_{\mathrm{L2sqr}}(p_1(\boldsymbol{x}), p_2(\boldsymbol{x}))d\boldsymbol{x}, \tag{A.8}$$

where

$$d'_{\mathrm{L2sqr}}(p_1(\boldsymbol{x}), p_2(\boldsymbol{x})) = (p_1(\boldsymbol{x}) - p_2(\boldsymbol{x}))^2. \tag{A.9}$$

In Fig. A.2, $d'_{\mathrm{L2sqr}}(p_1(\boldsymbol{x}), p_2(\boldsymbol{x}))$ is plotted as a function of $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$. Experimentally, using the L2 distance between Gaussian mixture models does not work well for genre classification. In unpublished nearest neighbor experiments on the ISMIR 2004 genre classification training set, we obtained 42% accuracy using the L2 distance compared to 65% using the symmetrized Kullback-Leibler divergence (in the experiments, nearest neighbor songs by the same artist as the query song were ignored). From this it would seem that the success of the symmetrized Kullback-Leibler divergence in music information retrieval is crucially linked to it asymptotically going towards infinity when one of p_1 and p_2 goes towards zero, i.e., it highly penalizes differences. This is supported by the observation in [10] that only a minority of a song's MFCCs actually discriminate it from other songs.

A disadvantage of using Gaussian mixture models to aggregate the MFCCs is that the temporal development of sounds is not taken into account, even though it is important to the perception of timbre [13, 14]. As noted in [10], a song can be modelled by the same Gaussian mixture model whether it is played forwards or backwards, even though it clearly makes an audible difference. Another disadvantage is that when two instruments play simultaneously, the probability density function (pdf) of the MFCCs will in general change rather unpredictably. If the two instruments only have little overlap in the mel-frequency domain, they will still be approximately linearly mixed after taking the logarithm in step 4 in Section 2.1 and after the discrete cosine transform, since the latter is a linear operation. However, the pdf of a sum of two stochastic variables is the convolution of the pdf of each of the variables. Only if the instruments do not play simultaneously will the resulting pdf contain separate peaks for each instrument. To

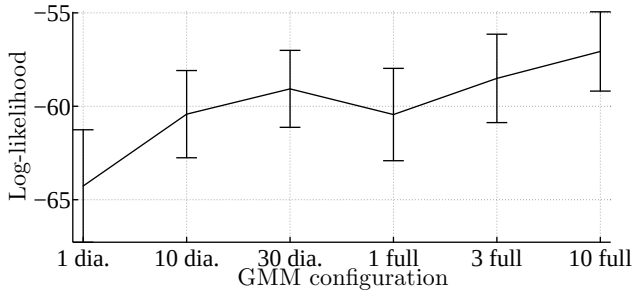


Fig. A.3: Log-likelihood for various Gaussian mixture model configurations. The number denotes the number of Gaussians in the mixture, and the letter is 'd' for diagonal covariance matrices and 'f' for full covariance matrices.

make matters even worse, such considerations also apply when chords are being played, and in this case it is almost guaranteed that some harmonics will fall into the same frequency bands, removing even the possibility of non-overlapping spectra.

With Gaussian mixture models, the covariance matrices are often assumed to be diagonal for computational simplicity. In [7, 8] it was shown that instead of a Gaussian mixture model where each Gaussian component has diagonal covariance matrix, a single Gaussian with full covariance matrix can be used without sacrificing discrimination performance. This simplifies both training and evaluation, since the closed form expressions in (A.2), (A.3) and (A.5) can be used. If the inverse of the covariance matrices are precomputed, (A.5) can be evaluated quite efficiently since the trace term only requires the diagonal elements of $(\boldsymbol{\Sigma}_1{}^{-1}\boldsymbol{\Sigma}_2)$ to be computed. For the symmetric version, the log terms even cancel, thus not even requiring the determinants to be precomputed. In Fig. A.3, the average log-likelihoods for 30 randomly selected songs from the ISMIR 2004 genre classification training database are shown for different Gaussian mixture model configurations. The figure shows that log-likelihoods for a mixture of 10 Gaussians with diagonal covariances and one Gaussian with full covariance matrix is quite similar. Using 30 Gaussians with diagonal covariance matrices increases the log-likelihood, but as shown in [9], genre classification performance does not benefit from this increased modelling accuracy. Log-likelihoods indicate only how well a model has captured the underlying density of the data, and not how well the models will discriminate in a classification task.

Table A.1: The six sound fonts used for the experiments.

Number	Sound font
1	AirFont 340 v1.01
2	Fluid R3 GM
3	GeneralUser GS 1.4
4	PersonalCopy 51f
5	RealFont 2.1
6	SGM-180 v1.5

3 Experiments

In this section, we present six experiments that further investigate the behavior of the MFCC-Gaussian-KL approach. The basic assumption behind all the experiments is that this approach is a timbral distance measure and that as such it is supposed to perform well at instrument classification. In all experiments we thus see how the instrument recognition performance is affected by various transformations and distortions. To perform the experiments, we take a number of MIDI files that are generated with Microsoft Music Producer and modify them in different ways to specifically show different MFCC properties. To synthesize wave signals from the MIDI files, we use the software synthesizer TiMidity++ version 2.13.2 with the six sound fonts listed in Table A.1. As each sound font uses different instrument samples, this approximates using six different realizations of each instrument. To compute MFCCs, we use the implementation in the Intelligent Sound Project toolbox that originates from the VOICEBOX toolbox by Mike Brookes. This implementation is described in [17] and includes frequencies up to 11025 Hz in the MFCCs. To aggregate the MFCCs from each synthesized MIDI file, we use the approach with a single Gaussian with full covariance matrix, since this would be the obvious choice in practical applications due to the clear computational advantages. All experiments have been performed with a number of different MFCC orders to see how it affects the results. We use a:b to denote MFCCs where the ath to the bth coefficient have been kept after the discrete cosine transform. As an example, 0:6 is where the DC coefficient and the following six coefficients have been kept. The experiments are implemented in MATLAB, and the source code, MIDI files and links to the sound fonts are available online [1].

[1] `http://kom.aau.dk/~jhj/publications/`

3.1 Timbre vs. Melody Classification

The first experiment is performed to verify that the MFCC-Gaussian-KL approach described in Section 2 also groups songs by instrumentation when an instrument plays several notes simultaneously. Due to the simple relation between harmonics in chords, the MFCC-Gaussian-KL approach could equally well match songs with similar chords than songs with identical instrumentation. When we refer to melodies in this section, we are thus not concerned with the lead melody, but rather with the chords and combinations of notes that are characteristic to a particular melody.

To perform the experiment, we take 30 MIDI songs of very different styles and the 30 MIDI instruments listed in Table A.2. For all combinations of songs and instruments, we perform the following:

1. Read MIDI song i.
2. Remove all percussion.
3. Force all notes to be played by instrument j.
4. Synthesize a wave signal $s_{ij}(n)$.
5. Extract MFCCs.
6. Train a multivariate Gaussian probability density function, $p_{ij}(\boldsymbol{x})$, on the MFCCs.

Next, we perform nearest neighbor classification on the $30 \times 30 = 900$ songs, i.e., for each song we compute:

$$(p, q) = \underset{\substack{k,l \\ (k,l)\neq(i,j)}}{\arg\min}\ d_{\mathrm{sKL}}(p_{ij}, p_{kl}). \tag{A.10}$$

If the nearest neighbor to song $s_{ij}(n)$, played with instrument j, is $s_{pq}(n)$, and it is also played with instrument j, i.e. $j = q$, then there is a match of instruments. We define the instrument classification rate by the fraction of songs where the instrument of a song and its nearest neighbor matches. Similarly, we define the melody classification rate by the fraction of songs where $i = p$. We repeat the experiment for the different sound fonts. Forcing all notes in a song to be played by the same instrument is not realistic, since e.g. the bass line would usually not be played with the same instrument as the main melody. However, using only the melody line would be an oversimplification. Keeping the percussion, which depends on the song, i, would also blur the results, although in informal experiments, keeping it only decreases the instrument classification accuracy by a few percentage points. In Fig. A.4, instrument and melody classification rates

are shown as a function of the MFCC order and the sound font used. From the figure, it is evident that when using even a moderate number of coefficients, the MFCC-Gaussian-KL approach is successful at identifying the instrument and is almost completely unaffected by the variations in the note and chord distributions present in the different songs.

3.2 Ensembles

Next, we repeat the experiment from the previous section using three different instruments for each song instead of just one. We select 30 MIDI files that each have three non-percussive tracks, and we select three sets with three instruments each. Let $\{a_1, a_2, a_3\}$, $\{b_1, b_2, b_3\}$ and $\{c_1, c_2, c_3\}$ denote the three sets, let $j, k, l \in 1, 2, 3$, and let i denote the MIDI file number. Similar to the experiment in Section 3.1, we perform the following for all combinations of i, j, k and l:

1. Read MIDI song i.
2. Remove all percussion.
3. Let all notes in the first, second and third track be played by instrument a_j, b_k and c_l, respectively.
4. Synthesize a wave signal $s_{ijkl}(n)$.
5. Extract MFCCs.
6. Train a multivariate Gaussian probability density function, $p_{ijkl}(\boldsymbol{x})$, on the MFCCs.

As before, the nearest neighbor is found, but this time according to

$$(p', q', r', s') = \underset{\substack{p,q,r,s \\ p \neq i}}{\arg\min}\, d_{\mathrm{sKL}}(p_{ijkl}, p_{pqrs}). \tag{A.11}$$

Thus, the nearest neighbor is not allowed to have the same melody as the query. This is to avoid that the nearest neighbor is the same melody with the instrument in a weak track replaced by another instrument. The fraction of nearest neighbors with the same three instruments, the fraction with at least two identical instruments and the fraction with at least one identical instrument is computed by counting how many of (q', r', s') equals (j, k, l).

In Fig. A.5, the fractions of nearest neighbors with different numbers of identical instruments are plotted. The fraction of nearest neighbors with two or more identical instruments is comparable to the instrument classification performance in Fig. A.4. To determine if the difficulties detecting all three instrument are caused by the MFCCs or the Gaussian model, we have repeated the experiments in Fig. A.6 with MFCCs 0:10 for the following seven setups:

Table A.2: The instruments used to synthesize the songs used for the experiments. All are from the General MIDI specification.

Number	Instrument name
1	Acoustic Grand Piano
11	Music Box
14	Xylophone
15	Tubular Bells
20	Church Organ
23	Harmonica
25	Acoustic Guitar (nylon)
37	Slap Bass 1
41	Violin
47	Orchestral Harp
53	Choir Aahs
54	Voice Oohs
57	Trumpet
66	Alto Sax
71	Bassoon
74	Flute
76	Pan Flute
77	Blown Bottle
79	Whistle
81	Lead 1 (square)
82	Lead 2 (sawtooth)
85	Lead 5 (charang)
89	Pad 1 (new age)
93	Pad 5 (bowed)
94	Pad 6 (metallic)
97	FX 1 (rain)
105	Sitar
110	Bag pipe
113	Tinkle Bell
115	Steel Drums

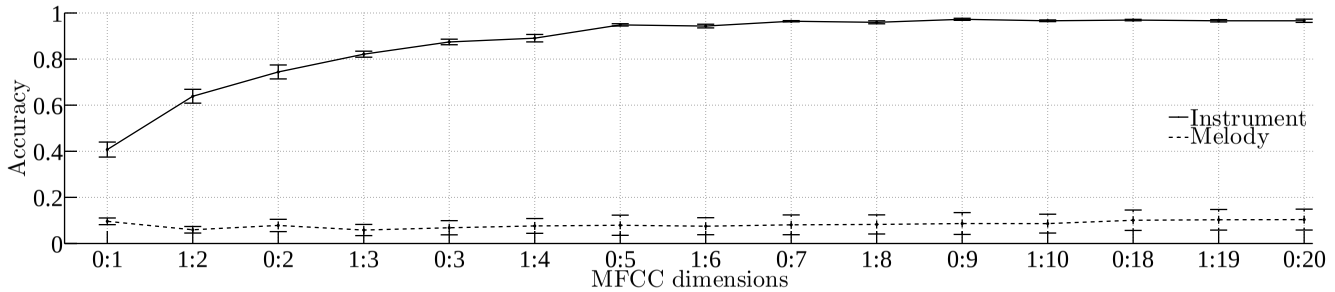


Fig. A.4: Mean and standard deviation of instrument and melody classification accuracies, i.e., the fraction of songs that have a song with the same instrumentation, or the same melody as nearest neighbor, respectively. For moderate MFCC orders, the instrument classification accuracy is consistently close to 1, and the melody classification accuracy is close to 0.

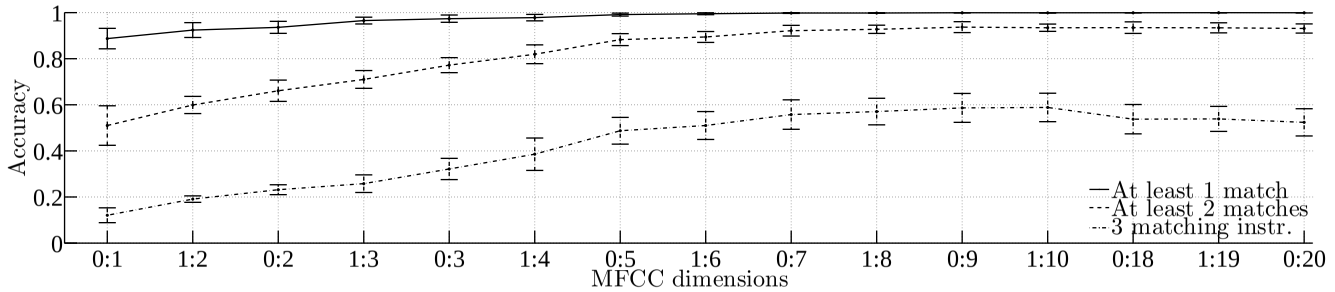


Fig. A.5: Mean and standard deviation of instrument classification accuracies when the success criterion is that the nearest neighbor has at least one, two or three identical instruments. Results are averaged over all six sound fonts.

- Using Gaussian mixture models with 10 and 30 diagonal covariance matrices, respectively.
- Gaussian mixture models with 1 and 3 full covariance matrices, respectively.
- Gaussian mixture models with 1 and 3 full covariance matrices, respectively, but where the instruments in a song are synthesized independently and subsequently concatenated into one song of triple length.
- Gaussian mixture models with 3 full covariance matrices where each instrument in a song is synthesized independently, and each Gaussian is trained on a single instrument only. The weights are set to $\frac{1}{3}$ each.
- Gaussian mixture model with 1 full covariance matrix, where, as a proof of concept, a non-negative matrix factorization (NMF) algorithm separates the MFCCs into individual sources that are concatenated before training the Gaussian model. The approach is a straightforward adoption of [35], where the NMF is performed between step 3 and 4 in the MFCC computation described in Section 2.1. As we, in line with [35], use a log-scale instead of the mel-scale, we should rightfully use the term LFCC instead of MFCC. Note that, like the first two setups, but unlike the setups based on independent instruments, this approach does not require access to the original, separate waveforms of each instrument, and thus is applicable to existing recordings.

From the additional experiments, it becomes clear that the difficulties capturing all three instruments originate from the simultaneous mixture. As we saw in Section 3.1, it does not matter that one instrument plays several notes at a time, but from Fig. A.5 and the "1 full add" experiment in Fig. A.6, we see that it clearly makes a difference whether different instruments play simultaneously. Although a slight improvement is observed when using separate Gaussians for each instrument, a single Gaussian actually seems to be adequate for modelling all instruments as long as different instruments do not play simultaneously. We also see that the NMF-based separation algorithm increases the number of cases where all three instruments are recognized. It conveniently simplifies the source separation task that a single Gaussian is sufficient to model all three instruments, since it eliminates the need to group the separated sources into individual instruments.

3.3 Different Realizations of the Same Instrument

In Section 3.1, we saw that the MFCC-Gaussian-KL approach was able to match songs played by the same instrument when they had been synthesized using the

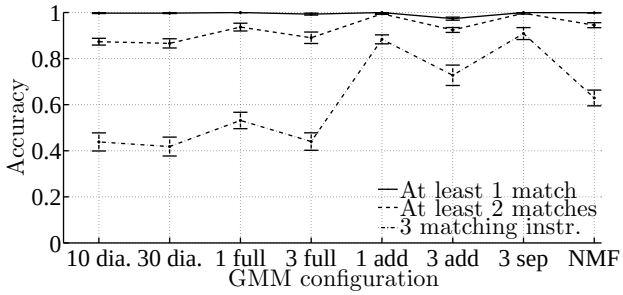


Fig. A.6: Instrument classification rates for different configurations of the Gaussian mixture model. The numbers denote the number of Gaussians in the mixture, and "dia." and "full" refer to the covariance matrices. For both "add" and "sep", each instrument has been synthesized independently. For "add", the tracks were concatenated to a single signal, while for "sep", the three equally weighted Gaussians were trained separately for each track. For "NMF", an NMF source separation algorithm has been applied. Results are averaged over all six sound fonts.

same sound font. In this section, to get an idea of how well this approach handles two different realizations of the same instrument, we use synthesized songs from different sound fonts as test and training data and measure the instrument classification performance once again. To the extent that a human listener would consider one instrument synthesized with two different sound fonts more similar than the same instrument synthesized by the first sound font and another instrument synthesized by the second, this experiment can also be considered a test of how well the MFCC-Gaussian-KL approach approximates human perception of timbral similarity.

The experimental setup is that of Section 3.1, only we use two different sound fonts, sf_m and sf_n, to synthesize two wave signals, $s_{ij}^{(\mathrm{sf}_m)}(n)$ and $s_{ij}^{(\mathrm{sf}_n)}(n)$, and estimate two multivariate Gaussians probability density functions, $p_{ij}^{(\mathrm{sf}_m)}(\boldsymbol{x})$ and $p_{ij}^{(\mathrm{sf}_n)}(\boldsymbol{x})$. We perform nearest neighbor classification again, but this time with a query synthesized with sf_m and a training set synthesized with sf_n, i.e., (A.10) is modified to

$$(p, q) = \underset{\substack{k,l \\ (k,l)\neq(i,j)}}{\arg\min}\ d_{\mathrm{sKL}}(p_{ij}^{(\mathrm{sf}_m)}, p_{kl}^{(\mathrm{sf}_n)}). \tag{A.12}$$

We test all combinations of the sound fonts mentioned in Table A.1. The resulting instrument classification rates are shown in Fig. A.7, and we see that the performance when using two different sound fonts are relatively low. We expect the low performance to have the same cause as the album effect [18]. In [36], the same phenomenon was observed when classifying instruments across different databases of real instrument sounds, and they significantly increased classification performance by using several databases as training set. However,

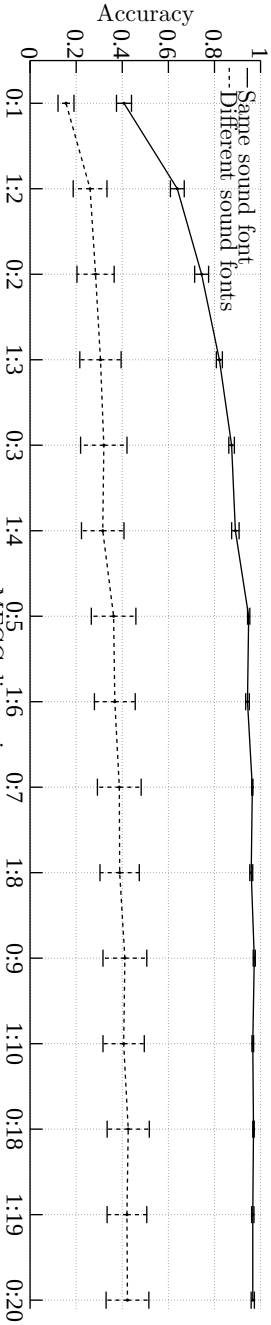


Fig. A.7: Mean and standard deviation of instrument classification accuracies when mixing different sound fonts.

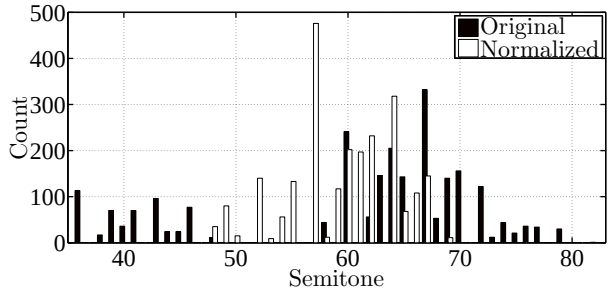


Fig. A.8: Histogram of notes in a MIDI song before and after normalization. The x-axis is the MIDI note number, i.e., 64 is middle C on the piano. The tonal range of the original song is much larger than that of the normalized song.

this is not directly applicable in our case, since the MFCC-Gaussian-KL is a song-level distance measure without an explicit training step.

When using songs synthesized from the same sound font for query and training, it is unimportant whether we increase the MFCC order by including the 0th coefficient or the next higher coefficient. However, we have noted that when combining different sound fonts, including the 0th MFCC at the cost of one of the higher coefficients has noticeable impact on performance. Unfortunately, since it is highly dependent on the choice of sound fonts if performance increases or decreases, an unambiguous conclusion cannot be drawn.

3.4 Transposition

When recognizing the instruments that are playing, a human listener is not particularly sensitive to transpositions of a few semitones. In this section, we experimentally evaluate how the MFCC-Gaussian-KL approach behaves in this respect. The experiment is built upon the same framework as the experiment in Section 3.1 and is performed as follows:

1. Repeat step 1–3 of the experiment in Section 3.1.
2. Normalize the track octaves (see below).
3. Transpose the song T_m semitones.
4. Synthesize wave signals $s_{ij}^{(T_m)}(n)$.
5. Extract MFCCs.
6. Train a multivariate Gaussian probability density function, $p_{ij}^{(T_m)}(\boldsymbol{x})$.

The octave normalization consists of transposing all tracks (e.g. bass and melody) such that the average note is as close to C4 (middle C on the piano) as possible, while only transposing the individual tracks an integer number of octaves relative to each other. The purpose is to reduce the tonal range of the songs. If the tonal range is too large, the majority of notes in a song and its transposed version will exist in both versions, hence blurring the results (see Fig. A.8). By only shifting the tracks an integer number of octaves relative to each other, we ensure that all harmonic relations between the tracks are kept. This time, the nearest neighbor is found as

$$(p,q) = \underset{\substack{k,l \\ (k,l)\neq(i,j)}}{\arg\min}\ d_{\mathrm{sKL}}(p_{ij}^{(T_m)}, p_{kl}^{(T_0)}). \tag{A.13}$$

That is, we search for the nearest neighbor to $p_{ij}^{(T_m)}(\boldsymbol{x})$ among the songs that have only been normalized but have not been transposed any further. The instrument and melody classification rates are computed for 11 different values of T_m that are linearly spaced between -24 and 24, which means that we maximally transpose songs two octaves up or down.

In Fig. A.9, instrument classification performance is plotted as a function of the number of semitones the query songs are transposed. Performance is hardly influenced by transposing songs ± 5 semitones. Transposing 10 semitones, which is almost an octave, noticeably affects results. Transposing ± 24 semitones severely reduces accuracy. In Fig. A.10, where instrument classification performance is plotted as a function of the MFCC order, we see that the instrument recognition accuracy generally increase with increasing MFCC order, stagnating around 10.

3.5 Bandwidth

Since songs in an actual music database may not all have equal sample rates, we examine the sensitivity of the MFCC-Gaussian-KL approach to downsampling, i.e., reducing the bandwidth. We both examine what happens if we mix songs with different bandwidths, and what happens if all songs have reduced, but identical bandwidth. Again we consider the MFCCs a timbral feature and use instrument classification performance as ground truth.

Mixing bandwidths

This experiment is very similar to the transposition experiment in Section 3.4, only we reduce the bandwidths of the songs instead of transposing them. Practically, we use the MATLAB resample function to downsample the wave signal to $2 \cdot BW$ and upsample it to 22 kHz again. The nearest neighbor instrument

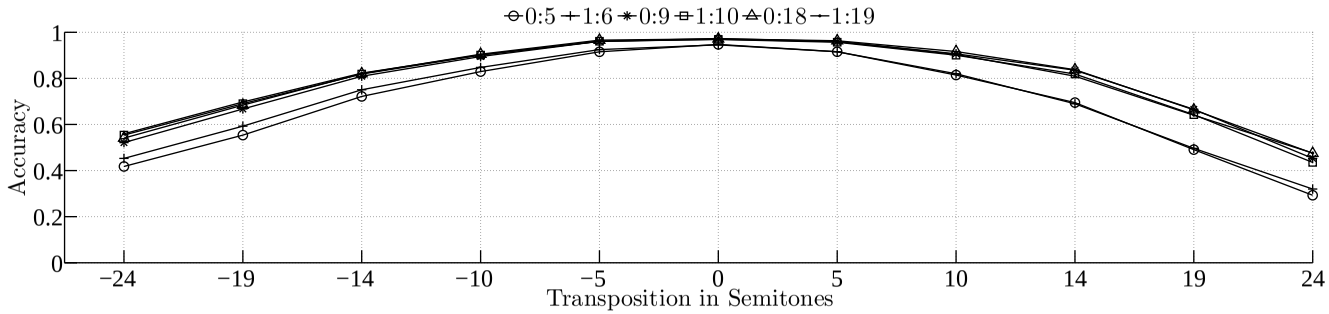


Fig. A.9: Instrument classification rate averaged over all sound fonts as a function of the number of semitones that query songs have been transposed.

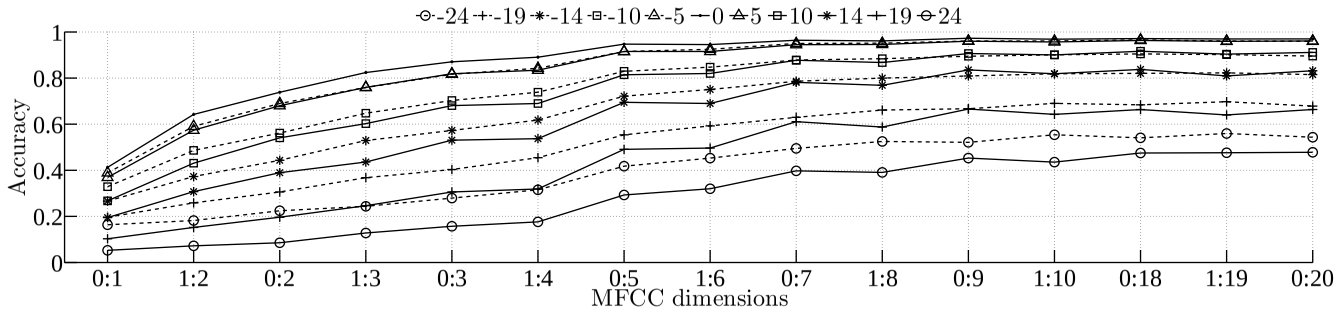


Fig. A.10: Instrument classification rate averaged over all sound fonts as a function of the number of MFCCs. The numbers -19, -14 etc. denote the number of semitones songs have been transposed.

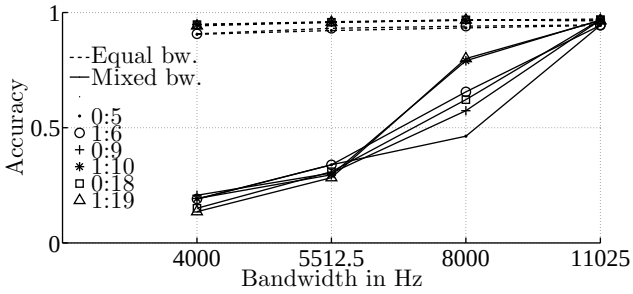


Fig. A.11: Average instrument classification accuracy averaged over all sound fonts when reducing the songs' bandwidths. For the mixed bandwidth results, the training set consists of songs with full bandwidth, while for the equal bandwidth results, songs in both the test and training sets have equal, reduced bandwidth.

classification rate is found as in (A.13) with $p_{ij}^{(T_m)}$ and $p_{kl}^{(T_0)}$ replaced by $p_{ij}^{(BW_m)}$ and $p_{kl}^{(BW_0)}$, respectively. The reference setting, BW_0, is 11 kHz, corresponding to a sampling frequency of 22 kHz.

Reducing bandwidth for all files

This experiment is performed as the experiment in Section 3.1, except that synthesized wave signals are downsampled to BW before computing the MFCCs for *both* test and training songs.

Results of both bandwidth experiments are shown in Fig. A.11. It is obvious from the figure that mixing songs with different bandwidths is a bad idea. Reducing the bandwidth of the query set from 11 kHz to 8 kHz significantly reduces performance, while reducing the bandwidth to 5.5 kHz, i.e., mixing sample rates of 22 kHz and 11 kHz, makes the distance measure practically useless with accuracies in the range from 30%–40%. On the contrary, if all songs have the same, low bandwidth, performance does not suffer significantly. It is thus clear that if different sampling frequencies can be encountered in a music collection, it is preferential to downsample all files to e.g. 8 kHz before computing the MFCCs. Since it is computationally cheaper to extract MFCCs from downsampled songs, and since classification accuracy is not noticeably affected by reducing the bandwidth, this might be preferential with homogeneous music collections as well. The experiment only included voiced instruments, so this result might not generalize to percussive instruments that often have more energy at high frequencies. In informal experiments on the ISMIR 2004 genre classification training database, genre classification accuracy only decreased by a few percentage points when downsampling all files to 8 kHz.

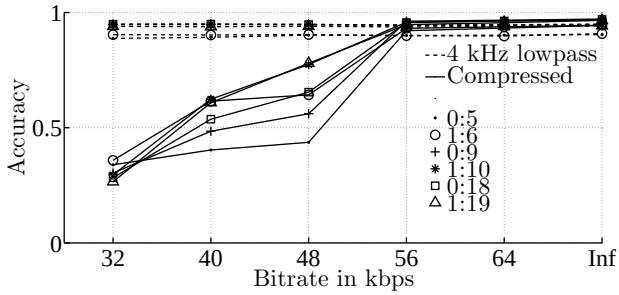


Fig. A.12: Instrument classification rates averaged over all sound fonts with MP3 compressed query songs as a function of bitrate.

3.6 Bitrate

Music is often stored in a compressed format. However, as shown in [17], MFCCs are sensitive to the spectral perturbations introduced by compression. In this section, we measure how these issues affect instrument classification performance. This experiment is performed in the same way as the transposition experiment in Section 3.4, except that transposing has been replaced by encoding to an MP3 file with bitrate B_m and decoding. Classification is also performed as given by (A.13). For MP3 encoding, the constant bitrate mode of LAME version 3.97 is used. The synthesized wave signal is in stereo when encoding but is converted to mono before computing the MFCCs. Results of different bitrates are shown in Fig. A.12. Furthermore, results of reducing the bandwidth to 4 kHz after decompression are also shown. Before compressing the wave signal, the MP3 encoder applies a lowpass filter. At 64 kbps, this lowpass filter has transition band from 10935 Hz to 11226 Hz, which is in the range of the very highest frequencies used when computing the MFCCs. Consequently, classification rates are virtually unaffected at a bitrate of 64 kbps. At 48 kbps, the transition band is between 7557 Hz and 7824 Hz, and at 32 kbps, the transition band is between 5484 Hz and 5677 Hz. The classification rates at 5.5 kHz and 8 kHz in Fig. A.11 and at 32 kbps and 48 kbps in Fig. A.12, respectively, are strikingly similar, hinting that bandwidth reduction is the major cause of the reduced accuracy. This is confirmed by the experiments where the bandwidth is always reduced to 4 kHz, which are unaffected by changing bitrates. So, if robustness to low bitrate MP3 encoding is desired, all songs should be downsampled before computing MFCCs.

4 Discussion

In all experiments, we let multivariate Gaussian distributions model the MFCCs from each song and used the symmetrized Kullback-Leibler divergence between the Gaussian distributions as distance measures. Strictly speaking, our results therefore only speak of the MFCCs with this particular distance measure and not of the MFCCs on their own. However, we see no obvious reasons that other classifiers would perform radically different.

In the first experiment, we saw that when keeping as little as four coefficients while excluding the 0th cepstral coefficient, instrument classification accuracy was above 80%. We therefore conclude that MFCCs primarily capture the spectral envelope when encountering a polyphonic mixture of voices from one instrument and not e.g. the particular structure encountered when playing harmonies.

When analyzing songs played by different instruments, only two of the three instruments were often recognized. The number of cases where all instruments were recognized increased dramatically when instruments were playing in turn instead of simultaneously, suggesting that the cause is either the log-step when computing the MFCCs, or the phenomenon that the probability density functions of a sum of random variables is the convolution of the individual probability density functions. From this it is clear that the success of the MFCC-Gaussian-KL approach in genre and artist classification is very possible due only to instrument/ensemble detection. This is supported by [37] that showed that for symbolic audio, instrument identification is very important to genre classification. We hypothesize that in genre classification experiments, recognizing the two most salient instruments is enough to achieve acceptable performance.

In the third experiment, we saw that the MFCC-Gaussian-KL approach does not consider songs with identical instrumentation synthesized with different sound fonts very similar. However, with non-synthetic music databases, e.g., [5] and [8], this distance measure seems to perform well even though different artists use different instruments. A possible explanation may be that the synthesized sounds are more homogeneous than a corresponding human performance, resulting in over-fitting of the multivariate Gaussian distributions. Another possibility is that what makes a real-world classifier work is the diversity among different performances in the training collection; i.e., if there are 50 piano songs in a collection, then a given piano piece may only be close to one or two of the other piano songs, while the rest, with respect to the distance measure, just as well could have been a trumpet piece or a xylophone piece. As observed in [8], performance of the MFCC-Gaussian-KL approach in genre classification increases significantly if songs by the same artist are in both the training and test collection, thus supporting the latter hypothesis. We speculate that relying more on the temporal development of sounds (for an example of this, see [38]) and less

on the spectral shape and using a more perceptually motivated distance measure instead of the Kullback-Leibler divergence can improve the generalization performance.

In [5] it is suggested that there is a "glass ceiling" for the MFCC-Gaussian-KL approach at 65%, meaning that no simple variation of it can exceed this accuracy. From the experiments, we can identify three possible causes of the glass ceiling:

1. The MFCC-Gaussian-KL approach neither takes melody nor harmony into account.
2. It is highly sensitive to different renditions of the same instrument.
3. It has problems identifying individual instruments in a mixture.

With respect to the second cause, techniques exists for suppressing channel effects in MFCC-based speaker identification. If individual instruments are separated in a preprocessing step, these techniques might be applicable to music as well. As shown in Section 3.2, a successful signal separation algorithm would also mitigate the third cause.

We measured the reduction in instrument classification rate when transposing songs. When transposing songs only a few semitones, instrument recognition performance was hardly affected, but transposing songs in the order of an octave or more causes performance to decrease significantly. When we compared MFCCs computed from songs with different bandwidths, we found that performance decreased dramatically. In contrast, if all songs had the same, low bandwidth, performance typically did not decrease more than 2–5 percentage points. Similarly, comparing MFCCs computed from low bitrate MP3 files and high bitrate files also affected instrument classification performance dramatically. The performance decrease for mixing bitrates matches the performance decrease when mixing bandwidths very well. If a song collection contains songs with different sample rates or different bitrates, it is recommended to downsample all files before computing the MFCCs.

5 Conclusion

We have analyzed the properties of a commonly used music similarity measure based on the Kullback-Leibler distance between Gaussian models of MFCC features. Our analyses show that the MFCC-Gaussian-KL measure of distance between songs recognizes instrumentation; a solo instrument playing several notes simultaneously does not degrade recognition accuracy, but an ensemble of instruments tend to suppress the weaker instruments. Furthermore, different realizations of instruments significantly reduces recognition performance. Our

results suggest that the use of source separation methods in combination with already existing music similarity measures may lead to increased classification performance.

Acknowledgment

The authors would like to thank Hans Laurberg for assistance with the non-negative matrix factorization algorithm used in Section 3.2.

References

[1] J. T. Foote, "Content-based retrieval of music and audio," in *Multimedia Storage and Archiving Systems II, Proc. of SPIE*, 1997, pp. 138–147.

[2] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proc. IEEE Int. Conf. Multimedia Expo*, Tokyo, Japan, 2001, pp. 745 – 748.

[3] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Processing*, vol. 10, pp. 293–301, 2002.

[4] J.-J. Aucouturier and F. Pachet, "Finding songs that sound the same," in *Proc. of IEEE Benelux Workshop on Model based Processing and Coding of Audio*, University of Leuven, Belgium, November 2002.

[5] ——, "Improving timbre similarity: How high's the sky?" *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.

[6] E. Pampalk, "Speeding up music similarity," in *2nd Annual Music Information Retrieval eXchange*, London, Sep. 2005.

[7] M. I. Mandel and D. P. W. Ellis, "Song-level features and support vector machines for music classification," in *Proc. Int. Symp. on Music Information Retrieval*, 2005, pp. 594–599.

[8] E. Pampalk, "Computational models of music similarity and their application to music information retrieval," Ph.D. dissertation, Vienna University of Technology, Austria, Mar. 2006.

[9] A. Flexer, "Statistical evaluation of music information retrieval experiments," Institute of Medical Cybernetics and Artificial Intelligence, Medical University of Vienna, Tech. Rep., 2005.

[10] J.-J. Aucouturier, "Ten experiments on the modelling of polyphonic timbre," Ph.D. dissertation, University of Paris 6, France, Jun. 2006.

[11] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl, "Aggregate features and AdaBoost for music classification," *Machine Learning*, vol. 65, no. 2–3, pp. 473–484, Dec. 2006.

[12] J. H. Jensen, M. G. Christensen, M. N. Murthi, and S. H. Jensen, "Evaluation of MFCC estimation techniques for music similarity," in *Proc. European Signal Processing Conf.*, Florence, Italy, 2006.

[13] T. D. Rossing, F. R. Moore, and P. A. Wheeler, *The Science of Sound*, 3rd ed. Addison-Wesley, 2002.

[14] B. C. J. Moore, *An introduction to the Psychology of Hearing*, 5th ed. Elsevier Academic Press, 2004.

[15] *Acoustical Terminology SI*, New York: American Standards Association Std., Rev. 1-1960, 1960.

[16] A. Nielsen, S. Sigurdsson, L. Hansen, and J. Arenas-Garcia, "On the relevance of spectral features for instrument classification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2007*, vol. 2, 2007, pp. II–485–II–488.

[17] S. Sigurdsson, K. B. Petersen, and T. Lehn-Schiøler, "Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music," in *Proc. Int. Symp. on Music Information Retrieval*, 2006.

[18] Y. E. Kim, D. S. Williamson, and S. Pilli, "Towards quantifying the "album effect" in artist identification," in *Proc. Int. Symp. on Music Information Retrieval*, 2006.

[19] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, pp. 357–366, Aug. 1980.

[20] A. V. Oppenheim and R. W. Schafer, "From frequency to quefrency: A history of the cepstrum," *IEEE Signal Processing Mag.*, vol. 21, pp. 95–106, Sep. 2004.

[21] ——, *Discrete-Time Signal Processing*, 1st ed. Prentice Hall, 1989.

[22] P. Stoica and N. Sandgren, "Smoothed nonparametric spectral estimation via cepsturm thresholding," *IEEE Signal Processing Mag.*, vol. 23, pp. 34–45, Nov. 2006.

[23] H. Terasawa, M. Slaney, and J. Berger, "Perceptual distance in timbre space," Limerick, Ireland, Jul. 2005.

[24] F. Zheng, G. Zhang, and Z. Song, "Comparison of different implementations of MFCC," *J. Computer Science & Technology*, vol. 16, pp. 582–589, Sep. 2001.

[25] M. Wölfel and J. McDonough, "Minimum variance distortionless response spectral estimation," *IEEE Signal Processing Mag.*, vol. 22, pp. 117 – 126, Sep. 2005.

[26] E. Pampalk, "A Matlab toolbox to compute music similarity from audio," in *Proc. Int. Symp. on Music Information Retrieval*, 2004, pp. 254–257.

[27] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. Int. Symp. on Music Information Retrieval*, 2000.

[28] Z. Liu and Q. Huang, "Content-based indexing and retrieval-by-example in audio," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2000, pp. 877 – 880.

[29] P. Stoica and R. Moses, *Spectral Analysis of Signals*. Prentice Hall, 2005.

[30] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, Dec. 1977.

[31] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood, and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, Apr. 1984.

[32] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. N. Y.: John Wiley & Sons, Inc., 1991.

[33] N. Vasconcelos, "On the complexity of probabilistic image retrieval," in *Proc. IEEE Int. Conf. Computer Vision*, Vancouver, BC, Canada, 2001, pp. 400–407.

[34] A. Berenzweig, "Anchors and hubs in audio-based music similarity," Ph.D. dissertation, Columbia University, New York, Dec. 2007.

[35] A. Holzapfel and Y. Stylianou, "Musical genre classification using nonnegative matrix factorization-based features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 424–434, Feb. 2008.

[36] A. Livshin and X. Rodet, "The importance of cross database evaluation in sound classification," in *Proc. Int. Symp. on Music Information Retrieval*, 2003.

[37] C. McKay and I. Fujinaga, "Automatic music classification and the importance of instrument identification," in *Proc. of the Conf. on Interdisciplinary Musicology*, 2005.

[38] A. Meng, P. Ahrendt, J. Larsen, and L. Hansen, "Temporal feature integration for music genre classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1654–1664, July 2007.

Paper B

Evaluation of Distance Measures Between Gaussian Mixture Models of MFCCs

Jesper Højvang Jensen, Daniel P.W. Ellis,
Mads Græsbøll Christensen, and Søren Holdt Jensen

This paper has been published in
Proceedings of the International Conference on Music Information Retrieval,
pp. 107–108, 2007.

The layout has been revised.

Abstract

In music similarity and in the related task of genre classification, a distance measure between Gaussian mixture models is frequently needed. We present a comparison of the Kullback-Leibler distance, the earth movers distance and the normalized L2 distance for this application. Although the normalized L2 distance was slightly inferior to the Kullback-Leibler distance with respect to classification performance, it has the advantage of obeying the triangle inequality, which allows for efficient searching.

1 Introduction

A common approach in computational music similarity is to extract mel-frequency cepstral coefficients (MFCCs) from a song, model them by a Gaussian mixture model (GMM) and use a distance measure between the GMMs as a measure of the musical distance between the songs [2, 3, 5]. Through the years, a number of distance measures between GMMs have been suggested, such as the Kullback-Leibler (KL) distance [2], optionally combined with the earth movers distance (EMD) [3]. In this article, we evaluate the performance of these two distance measures between GMMs together with the normalized L2 distance, which to our knowledge has not previously been used for this application.

2 Measuring Musical Distance

In the following, we shortly describe the Gaussian mixture model and the three distance measures between GMMs we have tested. Note that if a distance measure satisfies the triangle inequality, i.e., $d(p_1, p_3) \leq d(p_1, p_2) + d(p_2, p_3)$ for all values of $p1, p2$ and $p3$, then a nearest neighbor search can be speeded up by precomputing some distances. Assume we are searching for the nearest neighbor to p, and that we have just computed the distance to p_1. If we already know the distance between p_1 and p_2, then the distance to p_2 is bounded by $d(p, p_2) \geq d(p_1, p_2) - d(p_1, p)$. If the distance to the currently best candidate is smaller than $d(p_1, p_2) - d(p_1, p)$, we can discard p_2 without computing $d(p, p_2)$.

2.1 Gaussian Mixture Models

Due to intractability, the MFCCs extracted from a song are typically not stored but are instead modelled by a GMM. A GMM is a weighted sum of multivariate

Gaussians:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} c_k \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_k|}} \exp\Big(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\Big),$$

where K is the number of mixtures. For $K = 1$, a simple closed-form expression exists for the maximum-likelihood estimate of the parameters. For $K > 1$, the k-means algorithm and optionally the expectation-maximization algorithm are used to estimate the parameters.

2.2 Kullback-Leibler Distance

The KL distance is an information-theoretic distance measure between probability density functions. It is given by $d_{\mathrm{KL}}(p_1, p_2) = \int p_1(\boldsymbol{x}) \log \frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})} d\boldsymbol{x}$. As the KL distance is not symmetric, a symmetrized version, $d_{\mathrm{sKL}}(p_1, p_2) = d_{\mathrm{KL}}(p_1, p_2) + d_{\mathrm{KL}}(p_2, p_1)$, is usually used in music information retrieval. For Gaussian mixtures, a closed form expression for $d_{\mathrm{KL}}(p_1, p_2)$ only exists for $K = 1$. For $K > 1$, $d_{\mathrm{KL}}(p_1, p_2)$ is estimated using stochastic integration or the approximation in [4]. The KL distance does not obey the triangle inequality.

2.3 Earth Movers Distance

In this context the EMD is the minimum cost of changing one mixture into another when the cost of moving probability mass from component m in the first mixture to component n in the second mixture is given [3]. A common choice of cost is the symmetrized KL distance between the individual Gaussian components. With this cost, the EMD does not obey the triangle inequality.

2.4 Normalized L2 Distance

Let $p_i'(\boldsymbol{x}) = p_i(\boldsymbol{x}) / \sqrt{\int p_i(\boldsymbol{x})^2 d\boldsymbol{x}}$, i.e., $p_i(\boldsymbol{x})$ scaled to unit L2-norm. We then define the normalized L2 distance by $d_{\mathrm{nL2}}(p_1, p_2) = \int (p_1'(\boldsymbol{x}) - p_2'(\boldsymbol{x}))^2 d\boldsymbol{x}$. Since the ordinary L2 distance obeys the triangle inequality, and since we can simply prescale all GMMs to have unit L2-norm and then consider the ordinary L2 distance between the scaled GMMs, the normalized L2 distance will also obey the triangle inequality. Also note that $d_{\mathrm{nL2}}(p_1, p_2)$ is nothing but a continuous version of the cosine distance [6], since $d_{\mathrm{nL2}}(p_1, p_2) = 2(1 - \int p_1'(\boldsymbol{x}) p_2'(\boldsymbol{x}) d\boldsymbol{x})$. For GMMs, closed form expressions for the normalized L2 distance can be derived for any K from [1, Eq. (5.1) and (5.2)].

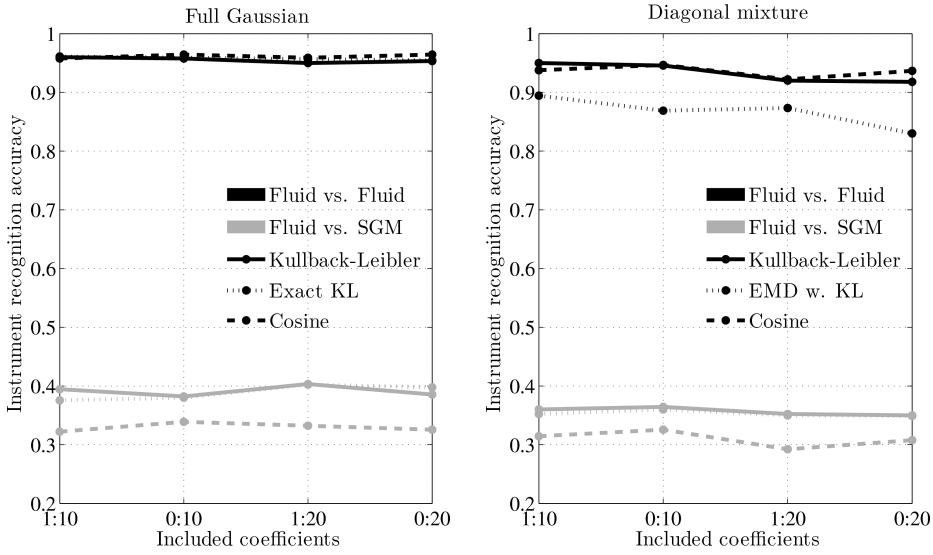


Fig. B.1: Instrument recognition results. Labels on the x-axis denotes the number of MFCCs retained, i.e. 0:10 means retaining the first 11 coefficients including the 0th. "Fluid" and "SGM" denotes the Fluid R3 and SGM 180 sound fonts, respectively.

3 Evaluation

We have evaluated the symmetrized KL distance computed by stochastic integration using 100 samples, EMD with the exact, symmetrized KL distance as cost, and the normalized L2 distance. We extract the MFCCs with the ISP toolbox R1 using default options[1]. To model the MFCCs we have both used a single Gaussian with full covariance matrix and a mixture of ten Gaussians with diagonal covariance matrices. With a single Gaussian, the EMD reduces to the exact, symmetrized KL distance. Furthermore, we have used different numbers of MFCCs. As the MFCCs are timbral features and therefore are expected to model instrumentation rather than melody or rhythm, we have evaluated the distance measures in a synthetic nearest neighbor instrument classification task using 900 synthesized MIDI songs with 30 different melodies and 30 different instruments. In Figure B.1, results for using a single sound font and results where the query song is synthesized by a different sound font than the songs it is compared to are shown. The former test can be considered a sanity test, and the latter test reflects generalization behaviour. Moreover, we have evaluated the distance measures using 30 s excerpts of the training songs from the MIREX

[1] `http://isound.kom.auc.dk/`

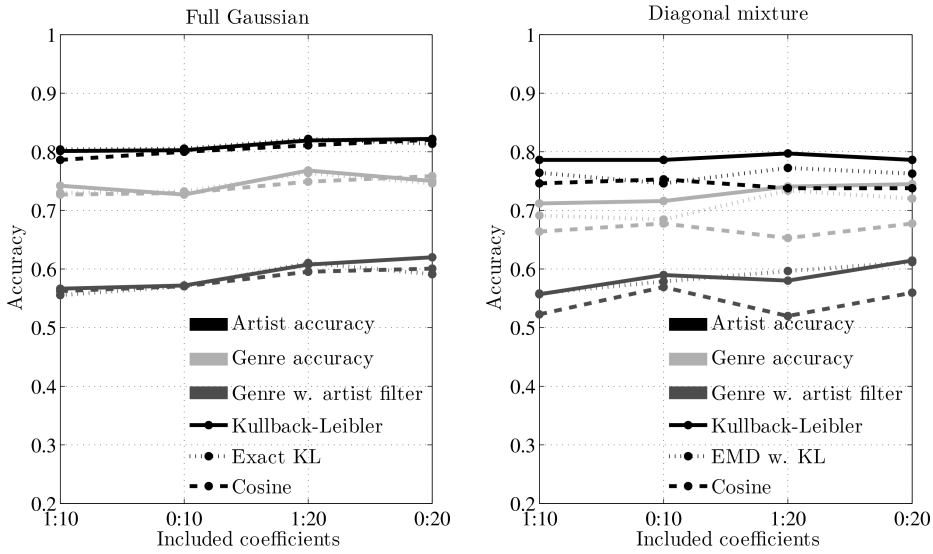


Fig. B.2: Genre and artist classification results for the MIREX 2004 database.

2004 genre classification contest, which consists of 729 songs from 6 genres. Results for genre classification, artist identification and genre classification with an artist filter (see [5]) are shown in Figure B.2.

4 Discussion

As the results show, all three distance measures perform approximately equal when using a single Gaussian with full covariance matrix, except that the normalized L2 distance performs a little worse when mixing instruments from different sound fonts. Using a mixture of ten diagonal Gaussians generally decrease recognition rates slightly, although it should be noted that [2] recommends using more than ten mixtures. For ten mixtures, the recognition rate for the Kullback-Leibler distance seems to decrease less than for the EMD and the normalized L2 distance. From these results we conclude that the cosine distance performs slightly worse than the Kullback-Leibler distance in terms of accuracy. However, with a single Gaussian having full covariance matrix this difference is negligible, and since the cosine distance obeys the triangle inequality, it might be preferable in applications with large datasets.

References

[1] P. Ahrendt, "The multivariate gaussian probability distribution," Technical University of Denmark, Tech. Rep., 2005.

[2] J.-J. Aucouturier, "Ten experiments on the modelling of polyphonic timbre," Ph.D. dissertation, University of Paris 6, France, Jun. 2006.

[3] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proc. IEEE Int. Conf. Multimedia Expo*, Tokyo, Japan, 2001, pp. 745 – 748.

[4] E. Pampalk, "Speeding up music similarity," in *2nd Annual Music Information Retrieval eXchange*, London, Sep. 2005.

[5] ——, "Computational models of music similarity and their application to music information retrieval," Ph.D. dissertation, Vienna University of Technology, Austria, Mar. 2006.

[6] J. R. Smith, "Integrated spatial and feature image systems: Retrieval, analysis and compression," Ph.D. dissertation, Columbia University, New York, 1997.

Paper C

Evaluation of MFCC Estimation Techniques for Music Similarity

Jesper Højvang Jensen, Mads Græsbøll Christensen,
Manohar N. Murthi, and Søren Holdt Jensen

This paper has been published in
Proceedings of the European Signal Processing Conference,
pp. 926–930, 2006.

The layout has been revised.

Abstract

Spectral envelope parameters in the form of mel-frequency cepstral coefficients are often used for capturing timbral information of music signals in connection with genre classification applications. In this paper, we evaluate mel-frequency cepstral coefficient (MFCC) estimation techniques, namely the classical FFT and linear prediction based implementations and an implementation based on the more recent MVDR spectral estimator. The performance of these methods are evaluated in genre classification using a probabilistic classifier based on Gaussian Mixture models. MFCCs based on fixed order, signal independent linear prediction and MVDR spectral estimators did not exhibit any statistically significant improvement over MFCCs based on the simpler FFT.

1 Introduction

Recently, the field of music similarity has received much attention. As people convert their music collections to mp3 and similar formats, and store thousands of songs on their personal computers, efficient tools for navigating these collections have become necessary. Most navigation tools are based on metadata, such as artist, album, title, etc. However, there is an increasing desire to browse audio collections in a more flexible way. A suitable distance measure based on the sampled audio signal would allow one to go beyond the limitations of human-provided metadata. A suitable distance measure should ideally capture instrumentation, vocal, melody, rhythm, etc. Since it is a non-trivial task to identify and quantify the instrumentation and vocal, a popular alternative is to capture the timbre [1–3]. Timbre is defined as "the auditory sensation in terms of which a listener can judge that two sounds with same loudness and pitch are dissimilar" [4]. The timbre is expected to depend heavily on the instrumentation and the vocals. In many cases, the timbre can be accurately characterized by the spectral envelope. Extracting the timbre is therefore similar to the problem of extracting the vocal tract transfer function in speech recognition. In both cases, the spectral envelope is to be estimated while minimizing the influence of individual sinusoids.

In speech recognition, mel-frequency cepstral coefficients (MFCCs) are a widespread method for describing the vocal tract transfer function [5]. Since timbre similarity and estimating the vocal tract transfer function are closely related, it is no surprise that MFCCs have also proven successful in the field of music similarity [1–3, 6, 7]. In calculating the MFCCs, it is necessary to estimate the magnitude spectrum of an audio frame. In the speech recognition community, it has been customary to use either fast Fourier transform (FFT) or linear prediction (LP) analysis to estimate the frequency spectrum. However, both

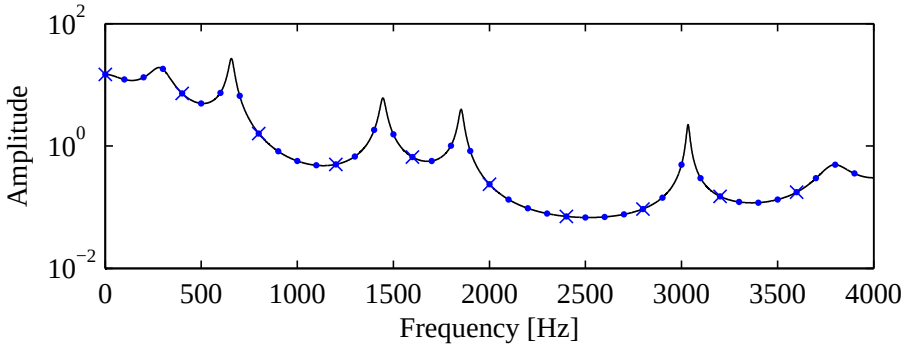


Fig. C.1: Spectrum of the signal that is excited by impulse trains in Figure C.3. Dots denote multiples of 100 Hz, and crosses denote multiples of 400 Hz.

methods do have some drawbacks. Minimum variance distortionless response (MVDR) spectral estimation has been proposed as an alternative to FFT and LP analysis [8, 9]. According to [10, 11], this increases speech recognition rates.

In this paper, we compare MVDR to FFT and LP analysis in the context of music similarity. For each song in a collection, MFCCs are computed and a Gaussian mixture model is trained. The models are used to estimate the genre of each song, assuming that similar songs share the same genre. We perform this for different spectrum estimators and evaluate their performance by the computed genre classification accuracies.

The outline of this paper is as follows. In Section 2, we summarize how MFCCs are calculated, what the shortcomings of the FFT and LP analysis as spectral estimators are, the idea of MVDR spectral estimation, and the advantage of prewarping. Section 3 describes how genre classification is used to evaluate the spectral estimation techniques. In Section 4, we present the results, and in Section 5, the conclusion is stated.

2 Spectral Estimation Techniques

In the following descriptions of spectrum estimators, the spectral envelope in Figure C.1 is taken as starting point. When a signal with this spectrum is excited by an impulse train, the spectrum becomes a line spectrum that is non-zero only at multiples of the fundamental frequency. The problem is to estimate the spectral envelope from the observed line spectrum. Before looking at spectrum estimation techniques, we briefly describe the application, i.e. estimation of mel-frequency cepstral coefficients.

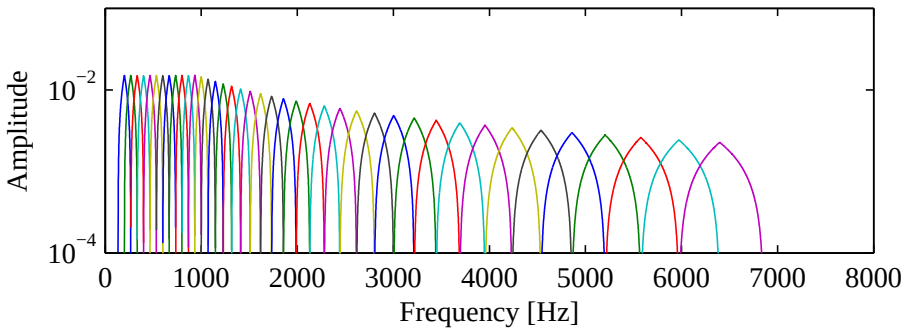


Fig. C.2: Mel bands

2.1 Mel-Frequency Cepstral Coefficients

Mel-frequency cepstral coefficients attempt to capture the perceptually most important parts of the spectral envelope of audio signals. They are calculated in the following way [12]:

1. Calculate the frequency spectrum
2. Filter the magnitude spectrum into a number of bands (40 bands are often used) according to the mel-scale, such that low frequencies are given more weight than high frequencies. In Figure C.2, the bandpass filters that are used in [12] are shown. We have used the same filters.
3. Sum the frequency contents of each band.
4. Take the logarithm of each sum.
5. Compute the discrete cosine transform (DCT) of the logarithms.

The first step reflects that the ear is fairly insensitive to phase information. The averaging in the second and third steps reflect the frequency selectivity of the human ear, and the fourth step simulates the perception of loudness. Unlike the other steps, the fifth step is not directly related to human sound perception, since its purpose is to decorrelate the inputs and reduce the dimensionality.

2.2 Fast Fourier Transform

The fast Fourier transform (FFT) is the Swiss army knife of digital signal processing. In the context of speech recognition, its caveat is that it does not attempt to suppress the effect of the fundamental frequency and the harmonics.

In Figure C.3, the magnitude of the FFT of a line spectrum based on the spectral envelope in Figure C.1 is shown. The problem is most apparent for high fundamental frequencies.

2.3 Linear Prediction Analysis

LP analysis finds the spectral envelope under the assumption that the excitation signal is white. For voiced speech with a high fundamental frequency, this is not a good approximation. Assume that $w(n)$ is white, wide sense stationary noise with unity variance that excites a filter having impulse response $h(n)$. Let $x(n)$ be the observed outcome of the process, i.e. $x(n) = w(n) * h(n)$ where $*$ denotes the convolution operator, and let $a_1,\ a_2,\ \ldots,\ a_K$ be the coefficients of the optimal least squares prediction filter. The prediction error, $y(n)$, is then given by

$$y(n) = x(n) - \sum_{k=1}^{K} a_k x(n-k). \tag{C.1}$$

Now, let $A(f)$ be the transfer function of the filter that produces $y(n)$ from $x(n)$, i.e.,

$$A(f) = 1 - \sum_{k=1}^{K} a_k e^{-i2\pi fk}. \tag{C.2}$$

Moreover, let $H(f)$ be the Fourier transform of $h(n)$, and let $S_x(f)$ and $S_y(f)$ be the power spectra of $x(n)$ and $y(n)$, respectively. Assuming $y(n)$ is approximately white with variance σ_y^2, i.e. $S_y(f) = \sigma_y^2$, it follows that

$$\begin{aligned} S_y(f) = \sigma_y^2 &= S_x(f)|A(f)|^2 \\ &= S_w(f)|H(f)|^2|A(f)|^2. \end{aligned} \tag{C.3}$$

Rearranging this, we get

$$\frac{\sigma_y^2}{|A(f)|^2} = S_w(f)|H(f)|^2. \tag{C.4}$$

The variables on the left side of Equation (C.4) can all be computed from the autocorrelation function. Thus, when the excitation signal is white with unity variance, i.e. $S_w(f) = 1$, LP analysis can be used to estimate the transfer function. Unfortunately, the excitation signal is often closer to an impulse train than to white noise. An impulse train with time period T has a spectrum which is an impulse train with period $1/T$. If the fundamental frequency is low, the assumption of a white excitation signal is good, because the impulses are closely spaced in the frequency domain. However, if the fundamental frequency is high, the linear predictor will tend to place zeros such that individual frequencies are

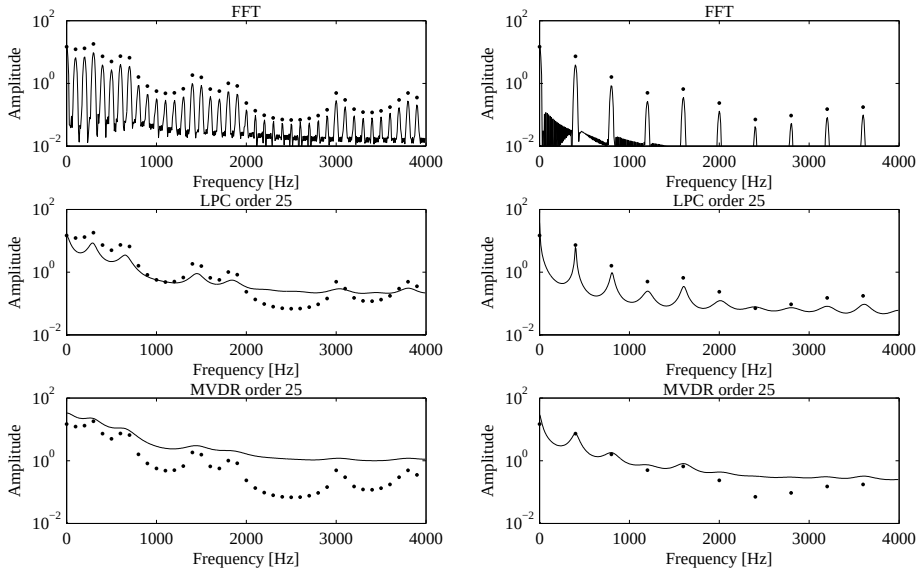


Fig. C.3: Three different spectral estimators. The dots denote the line spectres that can be observed from the input data. To the left, the fundamental frequency is 100 Hz, and to the right it is 400 Hz.

nulled, instead of approximating the inverse of the autoregressive filter $h(n)$. This is illustrated in Figure C.3, where two spectra with different fundamental frequencies have been estimated by LP analysis.

2.4 Minimum Variance Distortionless Response

Minimum variance distortionless response (MVDR) spectrum estimation has its roots in array processing [8, 9]. Conceptually, the idea is to design a filter $g(n)$ that minimizes the output power under the constraint that a specific frequency has unity gain. Let $\boldsymbol{R}_x$ be the autocorrelation matrix of a stochastic signal $x(n)$, and let $\mathbf{g}$ be a vector representation of $g(n)$. The expected output power of $x(n) * g(n)$ is then equal to $\mathbf{g}^{\mathrm{H}}\boldsymbol{R}_x\mathbf{g}$. Let f be the frequency at which we wish to estimate the power spectrum. Define a steering vector $\mathbf{b}$ as

$$\mathbf{b} = \begin{bmatrix} 1 & e^{-2\pi i f} & \dots & e^{-2\pi i K f} \end{bmatrix}^{\mathrm{T}}. \tag{C.5}$$

Compute $\mathbf{g}$ such that the power is minimized under the constraint that $\mathbf{g}$ has unity gain at the frequency f:

$$\mathbf{g} = \arg\min_{\mathbf{g}} \mathbf{g}^{\mathrm{H}}\boldsymbol{R}_x\mathbf{g} \quad \text{s.t.} \quad \mathbf{b}^{\mathrm{H}}\mathbf{g} = 1. \tag{C.6}$$

The estimated spectral contents, $\widehat{S}_x(f)$, is then given by the output power of $x(n) * g(n)$:

$$\widehat{S}_x(f) = \mathbf{g}^{\mathrm{H}} \boldsymbol{R}_x \mathbf{g}. \tag{C.7}$$

It turns out that (C.6) and (C.7) can be reduced to the following expression [8, 9]:

$$\widehat{S}_x(f) = \frac{1}{\mathbf{b}^{\mathrm{H}} \boldsymbol{R}_x^{-1} \mathbf{b}}. \tag{C.8}$$

In Figure C.3, the spectral envelope is estimated using the MVDR technique. Compared to LP analysis with the same model order, the MVDR spectral estimate will be much smoother [13]. In MVDR spectrum estimation, the model order should ideally be chosen such that the filter is able to cancel all but one sinusoid. If the model order is significantly higher, the valleys between the harmonics will start to appear, and if the model order is lower, the bias will be higher [13]. It was reported in [11] that improvements in speech recognition had been obtained by using variable order MVDR. Since it is non-trivial to adapt their approach to music, and since [11] and [14] also have reported improvements with a fixed model order, we use a fixed model order in this work. Using a variable model order with music is a topic of current research.

2.5 Prewarping

All the three spectral estimators described above have in common that they operate on a linear frequency scale. The mel-scale, however, is approximately linear at low frequencies and logarithmic at high frequencies. This means that the mel-scale has much higher frequency resolution at low frequencies than at high frequencies. Prewarping is a technique for approximating a logarithmic frequency scale. It works by replacing all delay elements $z^{-1} = e^{-2\pi i f}$ by the all-pass filter

$$\tilde{z}^{-1} = \frac{e^{-2\pi i f} - \alpha}{1 - \alpha e^{-2\pi i f}}. \tag{C.9}$$

For a warping parameter $\alpha = 0$, the all-pass filter reduces to an ordinary delay. If α is chosen appropriately, then the warped frequency axis can be a fair approximation to the mel-scale [10, 11]. Prewarping can be applied to both LP analysis and MVDR spectral estimation [10, 11].

3 Genre Classification

The considerations above are all relevant to speech recognition. Consequently, the use of MVDR for spectrum estimation has increased speech recognition rates [11, 14, 15]. However, it is not obvious whether the same considerations hold for

music similarity. For instance, in speech there is only one excitation signal, while in music there may be an excitation signal and a filter for each instrument. In the following we therefore investigate whether MVDR spectrum estimation leads to an improved music similarity measure. Evaluating a music similarity measure directly involves numerous user experiments. Although other means of testing have been proposed, e.g. [16], genre classification is an easy, meaningful method for evaluating music similarity [7, 17]. The underlying assumption is that songs from the same genre are musically similar. For the evaluation, we use the training data from the ISMIR 2004 genre classification contest [18], which contains 729 songs that are classified into 6 genres: classical (320 songs, 40 artists), electronic (115 songs, 30 artists), jazz/blues (26 songs, 5 artists), metal/punk (45 songs, 8 artists), rock/pop (101 songs, 26 artists) and world (122 songs, 19 artists). Inspired by [2] and [3], we perform the following for each song:

1. Extract the MFCCs in windows of 23.2 ms with an overlap of 11.6 ms. Store the first eight coefficients.

2. Train a Gaussian mixture model with 10 mixtures and diagonal covariance matrices.

3. Compute the distance between all combinations of songs.

4. Perform nearest neighbor classification by assuming a song has the same genre as the most similar song apart from itself (and optionally apart from songs by the same artist).

We now define the accuracy as the fraction of correctly classified songs. The MFCCs are calculated in many different ways. They are calculated with different spectral estimators: FFT, LP analysis, warped LP analysis, MVDR, and warped MVDR. Except for the FFT, all spectrum estimators have been evaluated with different model orders. The non-warped methods have been tested both with and without the use of a Hamming window. For the warped estimators, the autocorrelation has been estimated as in [11]. Before calculating MFCCs, pre-filtering is often applied. In speech processing, pre-filtering is performed to cancel a pole in the excitation signal, which is not completely white as otherwise assumed [5]. In music, a similar line of reasoning cannot be applied since the excitation signal is not as well-defined as in speech due to the diversity of musical instruments. We therefore calculate MFCCs both with and without pre-filtering.

The Gaussian mixture model (GMM) for song l is given by

$$p_l(\mathbf{x}) = \sum_{k=1}^{K} c_k \frac{1}{\sqrt{|2\pi\mathbf{\Sigma}_k|}} \exp\Big(-\tfrac{1}{2}(\mathbf{x}-\mu_k)^{\mathrm{T}}\mathbf{\Sigma}_k^{-1}(\mathbf{x}-\mu_k)\Big), \tag{C.10}$$

where K is the number of mixtures. The parameters of the GMM, $\mu_1, \ldots, \mu_K$ and $\mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_K$, are computed with the k-means-algorithm. The centroids computed with the k-means-algorithm are used as means for the Gaussian mixture components, and the data in the corresponding Voronoi regions are used to compute the covariance matrices. This is often used to initialize the EM-algorithm, which then refines the parameters, but according to [16], and our own experience, there is no significant improvement by subsequent use of the EM-algorithm. As distance measure between two songs, an estimate of the symmetrized Kullback-Leibler distance between the Gaussian mixture models is used. Let $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ be the GMMs of two songs, and let $\mathbf{x}_{1\,1}, \ldots, \mathbf{x}_{1\,N}$ and $\mathbf{x}_{2\,1}, \ldots, \mathbf{x}_{2\,N}$ be random vectors drawn from $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$, respectively. We then compute the distance as in [3]:

$$d = \sum_{n=1}^{N} \big(\log(p_1(\mathbf{x}_{1\,n})) + \log(p_2(\mathbf{x}_{2\,n})) - \log(p_1(\mathbf{x}_{2\,n})) - \log(p_2(\mathbf{x}_{1\,n})) \big). \tag{C.11}$$

In our case, we set $N = 200$. When generating the random vectors, we ignore mixtures with weights $c_k < 0.01$ (but not when evaluating equation (C.11)). This is to ensure that outliers do not influence the result too much. When classifying a song, we either find the most similar song or the most similar song by another artist. According to [2, 7], this has great impact on the classification accuracy. When the most similar song is allowed to be of the same artist, artist identification is performed instead of genre classification.

4 Results

The computed classification accuracies are shown graphically in Figure C.4. When the most similar song is allowed to be of the same artist, i.e. songs of the same artist are included in the training set, accuracies are around 80%, and for the case when the same artist is excluded from the training set, accuracies are around 60%. This is consistent with [2], which used the same data set. With a confidence interval of 95%, we are not able to conclude that the fixed order MVDR and LP based methods perform better than the FFT-based methods.

In terms of complexity, the FFT is the winner in most cases. When the model order of the other methods gets high, the calculation of the autocorrelation function is done most efficiently by FFTs. Since this requires both an FFT and an inverse FFT, the LPC and MVDR methods will in most cases be computationally more complex than using the FFT for spectrum estimation. Furthermore, if the autocorrelation matrix is ill-conditioned, the standard Levinson-Durbin algorithm fails, and another approach, such as the pseudoinverse, must be used.

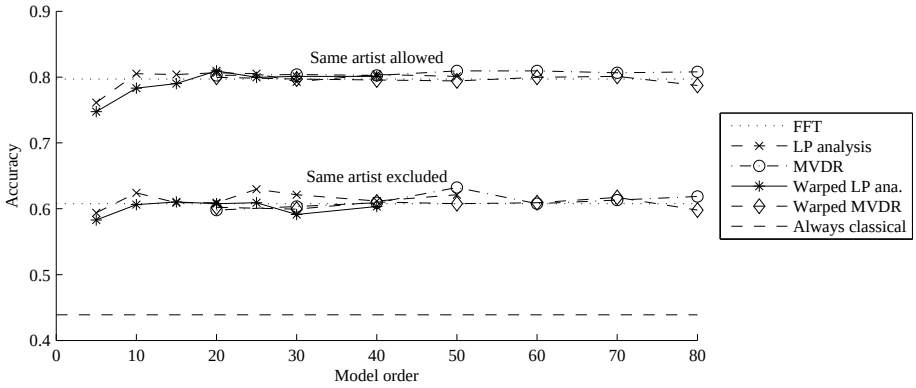


Fig. C.4: Classification accuracies. All methods are using preemphasis. The FFT, LP analysis and MVDR methods use a Hamming window.

The experiments have been performed both with and without a preemphasis filter. When allowing the most similar song to be of the same artist, a preemphasis filter increased accuracy in 43 out of 46 cases, and it decreased performance in two cases. When excluding the same artist, a preemphasis filter always increased accuracy. Of the total of 103 cases where performance was increased, the 37 were statistically significant with a 95% confidence interval.

The improvement by using a Hamming window depends on the spectral estimator. We restrict ourselves to only consider the case with a preemphasis filter, since this practically always resulted in higher accuracies. For this case, we observed that a Hamming window is beneficial in all tests but one test using the LPC and two using MVDR. In eight of the cases with an increase in performance, the result was statistically significant with a 95% confidence interval.

5 Conclusion

With MFCCs based on fixed order, signal independent LPC, warped LPC, MVDR, or warped MVDR, genre classification tests did not exhibit any statistically significant improvements over FFT-based methods. This means that a potential difference must be minor. Since the other spectral estimators are computationally more complex than the FFT, the FFT is preferable in music similarity applications. There are at least three possible explanations why the results are not statistically significant:

1. The choice of spectral estimator is not important.

2. The test set is too small to show subtle differences.

3. The method of testing is not able to reveal the differences.

The underlying reason is probably a combination of all three. When averaging the spectral contents of each mel-band (see Figure C.2), the advantage of the MVDR might be evened out. Although the test set consists of 729 songs, this does not ensure finding statistically significant results. Many of the songs are easily classifiable by all spectrum estimation methods, and some songs are impossible to classify correctly with spectral characteristics only. This might leave only a few songs that actually depend on the spectral envelope estimation technique. The reason behind the third possibility is that there is not a one-to-one correspondence between timbre, spectral envelope and genre. This uncertainty might render the better spectral envelope estimates useless.

References

[1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Processing*, vol. 10, pp. 293–301, 2002.

[2] A. Flexer, "Statistical evaluation of music information retrieval experiments," Institute of Medical Cybernetics and Artificial Intelligence, Medical University of Vienna, Tech. Rep., 2005.

[3] J.-J. Aucouturier and F. Pachet, "Improving timbre similarity: How high's the sky?" *Journal of Negative Results in Speech and Audio Sciences*, 2004.

[4] B. C. J. Moore, *An introduction to the Psychology of Hearing*, 5th ed. Elsevier Academic Press, 2004.

[5] J. John R. Deller, J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*, 2nd ed. Wiley-IEEE Press, 1999.

[6] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proc. IEEE International Conference on Multimedia and Expo*, Tokyo, Japan, 2001.

[7] E. Pampalk, "Computational models of music similarity and their application to music information retrieval," Ph.D. dissertation, Vienna University of Technology, Austria, Mar. 2006.

[8] M. N. Murthi and B. Rao, "Minimum variance distortionless response (MVDR) modeling of voiced speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Munich, Germany, April 1997.

[9] M. N. Murthi and B. D. Rao, "All-pole modeling of speech based on the minimum variance distortionless response spectrum," *IEEE Trans. Speech and Audio Processing*, vol. 8, no. 3, May 2000.

[10] M. Wölfel, J. McDonough, and A. Waibel, "Warping and scaling of the minimum variance distortionless response," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, November 2003, pp. 387 – 392.

[11] M. Wölfel and J. McDonough, "Minimum variance distortionless response spectral estimation," *IEEE Signal Processing Mag.*, vol. 22, pp. 117 – 126, Sept. 2005.

[12] M. Slaney, "Auditory toolbox version 2," Interval Research Corporation, Tech. Rep., 1998.

[13] M. N. Murthi, "All-pole spectral envelope modeling of speech," Ph.D. dissertation, University of California, San Diego, 1999.

[14] U. H. Yapanel and J. H. L. Hansen, "A new perspective on feature extraction for robust in-vehicle speech recognition," in *European Conf. on Speech Communication and Technology*, 2003.

[15] S. Dharanipragada and B. D. Rao, "MVDR-based feature extraction for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2001.

[16] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman, "A large-scale evaluation of acoustic and subjective music similarity measures," in *Proc. Int. Symp. on Music Information Retrieval*, 2003.

[17] T. Li and G. Tzanetakis, "Factors in automatic musical genre classificatin of audio signals," in *Proc. IEEE Workshop on Appl. of Signal Process. to Aud. and Acoust.*, 2003.

[18] ISMIR 2004 audio description contest – genre/artist ID classification and artist similarity. [Online]. Available: \url{http://ismir2004.ismir.net/genre_contest/index.htm}

Paper D

A Tempo-insensitive Distance Measure for Cover Song Identification based on Chroma Features

Jesper Højvang Jensen, Mads Græsbøll Christensen,
Daniel P. W. Ellis, and Søren Holdt Jensen

This paper has been published in
Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 2209–2212, 2008.

The layout has been revised.

Abstract

We present a distance measure between audio files designed to identify cover songs, which are new renditions of previously recorded songs. For each song we compute the chromagram, remove phase information and apply exponentially distributed bands in order to obtain a feature matrix that compactly describes a song and is insensitive to changes in instrumentation, tempo and time shifts. As distance between two songs, we use the Frobenius norm of the difference between their feature matrices normalized to unit norm. When computing the distance, we take possible transpositions into account. In a test collection of 80 songs with two versions of each, 38% of the covers were identified. The system was also evaluated on an independent, international evaluation where it despite having much lower complexity performed on par with the winner of last year.

1 Introduction

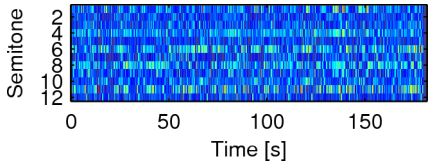
As the size of digital music collections increases, navigating such collections become increasingly difficult. One purpose of music information retrieval is to develop algorithms to facilitate such navigation, for instance by finding songs with similar instrumentation, rhythm or melody. Based on the initial success using MFCCs for genre classification, much research has until now directly or indirectly focused on finding songs with similar instrumentation [1–4]. With the introduction of a cover song identification contest in 2006, the Music Information Retrieval Evaluation eXchange (MIREX) community has put focus on musical structure rather than spectral statistics. In the MIREX 2006 cover song identification contest, the system in [5] had the best retrieval performance. This system had relatively high storage and computational requirements. It combines the chromagram, which is an octave-independent magnitude spectrum, with a beat tracker in order to obtain a beat-synchronous chromagram that is insensitive to differences in tempo.

Most cover song identification systems depend on estimates of musical properties and are therefore sensitive to the accuracy of the estimates. The system in [5] uses a beat estimate, [6] extracts the melody, and both [7] and [8] rely on chord recognition. Like [5, 7, 8], the proposed system is based on the chromagram, but unlike the aforementioned systems, it does not directly attempt to extract musical properties. Instead, it applies a number of transformations in order to obtain a feature that compactly describes a song and is not sensitive to instrumentation, time alignment or tempo. The feature is somewhat similar to the rhythm patterns in [9] that describe the amount of modulation in certain frequency bands, and the result is a system with performance similar to [5], but with a complexity that is heavily reduced.

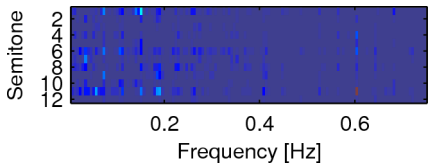
In Section 2 and 3, we describe the extracted features and the distance measure between them, respectively. We evaluate the performance of the proposed system in Section 4 before giving the conclusion in Section 5.

2 Feature extraction

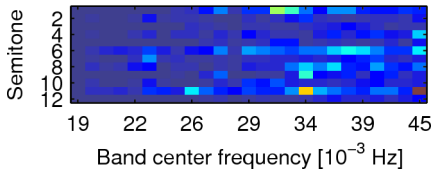
The assumptions behind the proposed system are that a song and its cover versions share the same melody, but might differ with respect to instrumentation, time shifts, tempo and transpositions. We extract a feature matrix which is insensitive to the former three properties, while the distance computation ensures invariance to transpositions. In Fig. D.1 and D.2, examples of a signal at different stages during the feature extraction are given, and in Fig. D.3 a block diagram of the process is shown. Note that except for a horizontal shift of one band, Fig. D.1(c) and D.2(c) are very similar.



(a) Chromagram after the logarithm.

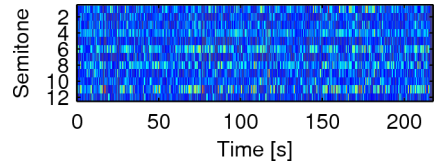


(b) Power spectrum of the chromagram rows.

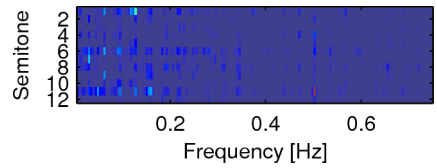


(c) Energy in the 25 exponentially spaced bands.

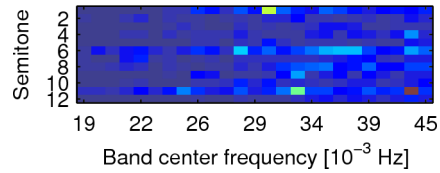
Fig. D.1: Different stages of feature extraction from a MIDI song with duration 3:02.



(a) Chromagram after the logarithm.



(b) Power spectrum of the chromagram rows.



(c) Energy in the 25 exponentially spaced bands.

Fig. D.2: Feature extraction from the same MIDI song as in Fig. D.1, except it is stretched to have duration 3:38.

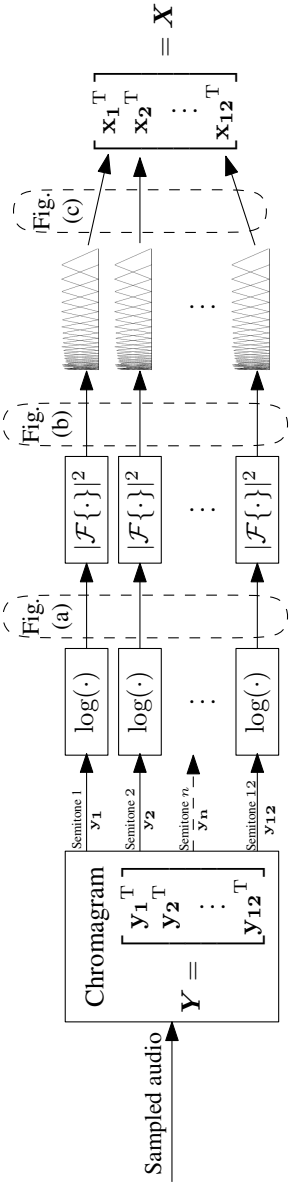


Fig. D.3: Block diagram of the feature extraction process.

The first stage of extracting the feature matrix is to compute the chromagram from a song. It is conceptually a short time spectrum which has been folded into a single octave [10]. This single octave is divided into 12 logarithmically spaced frequency bins that each correspond to one semitone on the western musical scale. Ideally, the chromagram would be independent of instrumentation and only reflect the notes of the music being played. We use the implementation described in [5] to compute the chromagram. We found that elementwise taking the logarithm of the chromagram increased performance, possibly because it better reflects human loudness perception. Let the chromagram matrix $\boldsymbol{Y}$ be given by

$$\boldsymbol{Y} = \begin{bmatrix} \mathbf{y}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{y}_{12}^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} y_1(1) & y_1(2) & \cdots & y_1(N) \\ & \vdots & & \\ y_{12}(1) & y_{12}(2) & \cdots & y_{12}(N) \end{bmatrix} \tag{D.1}$$

where $y_n(m)$ represents the magnitude of semitone n at frame m. The chromagram after the logarithm operation, $\boldsymbol{Y}_{\log} = [\mathbf{y}'_1, \cdots, \mathbf{y}'_{12}]^{\mathrm{T}}$, is given by $(\boldsymbol{Y}_{\log})_{i,j} = \log(1 + (\boldsymbol{Y})_{i,j}/\delta)$, where $(\cdot)_{i,j}$ is the element of row i and column j, and δ is a small constant.

To avoid time alignment problems, we remove all phase information from $\boldsymbol{Y}_{\log}$ by computing the power spectrum for each row, i.e.,

$$\boldsymbol{Y}_{\mathrm{pwr}} = \begin{bmatrix} |\mathcal{F}\{{\mathbf{y}'_1}^{\mathrm{T}}\}|^2 \\ \vdots \\ |\mathcal{F}\{{\mathbf{y}'_{12}}^{\mathrm{T}}\}|^2 \end{bmatrix}, \tag{D.2}$$

where $\mathcal{F}$ is the Fourier operator. This also removes all semitone co-occurence information, which may contain useful information.

Moving on to temporal differences, let $x(t)$ be a continuous signal and let $X(f) = \mathcal{F}\{x(t)\}$ be its Fourier transform. A temporal scaling of $x(t)$ will also cause a scaling in the frequency domain: $\mathcal{F}\{x(kt)\} = X(f/k)$. This approximately holds for discrete signals as well and thus for the rows of $\boldsymbol{Y}_{\mathrm{pwr}}$. For cover songs it is reasonable to assume that the ratio between the tempo of a song and its cover is bounded, i.e., that two songs do not differ in tempo more than, e.g., a factor c, in which case $\frac{1}{c} \leq k \leq c$. Now, if either the time or frequency axis is viewed on a logarithmic scale, a scaling (i.e., $k \neq 1$) will show up as an offset. This is used in e.g. [11] to obtain a representation where the distances between the fundamental frequency and its harmonics are independent of the fundamental frequency itself. If the scaling k is bounded, then the offset will be bounded as well. Thus, by sampling the rows of $\boldsymbol{Y}_{\mathrm{pwr}}$ on a logarithmic scale, we convert differences in tempo to differences in offsets. We implement this by representing each row of $\boldsymbol{Y}_{\mathrm{pwr}}$ by the output of a number of exponentially spaced bands. In Fig. D.4, the 25 bands with 50% overlap that we used are shown.

The lowest band start at 0.017 Hz, and the highest band end at 0.667 Hz, thus capturing variations on a time scale between 1.5 s and 60 s. The amount of temporal scaling allowed is further increased when computing the distance. The resulting feature is a 12×25 matrix where component i, j reflects the amount of modulation of semitone i in frequency band j. In comparison, if a song is 4 minutes long and has a tempo of 120 beats per minute, the beat-synchronous feature in [5] will have a dimension of 12×480.

3 Distance measure

We compute the distance between two feature matrices $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ by normalizing them to unit norm and compute the minimum Frobenius distance when allowing transpositions and frequency shifts. First, we normalize to unit Frobenius norm:

$$\boldsymbol{X}'_1 = \boldsymbol{X}_1 / \|\boldsymbol{X}_1\|_{\mathrm{F}}, \tag{D.3}$$

$$\boldsymbol{X}'_2 = \boldsymbol{X}_2 / \|\boldsymbol{X}_2\|_{\mathrm{F}}. \tag{D.4}$$

Let $\mathbf{T}_{12}$ be the 12×12 permutation matrix that transposes $\boldsymbol{X}'_1$ or $\boldsymbol{X}'_2$ by one semitone:

$$(\mathbf{T}_{12})_{i,j} = \begin{cases} (\boldsymbol{I})_{i+1,j} & \text{for } i < 12, \\ (\boldsymbol{I})_{1,j} & \text{for } i = 12, \end{cases} \tag{D.5}$$

where $\boldsymbol{I}$ is the identity matrix. To compensate for transpositions, we minimize the Frobenius distance over all possible transpositions:

$$d'(\boldsymbol{X}'_1, \boldsymbol{X}'_2) = \min_{p \in \{1,2,\cdots,12\}} \|\mathbf{T}_{12}^p \boldsymbol{X}'_1 - \boldsymbol{X}'_2\|_{\mathrm{F}}. \tag{D.6}$$

To allow even further time scaling than permitted by the effective bandwidths, we also allow shifting the matrices by up to two columns:

$$d(\boldsymbol{X}'_1, \boldsymbol{X}'_2) = \min_{s \in \{-2,-1,0,1,2\}} d'(\boldsymbol{X}'^{(s)}_1, \boldsymbol{X}'^{(-s)}_2), \tag{D.7}$$

where

$$\boldsymbol{X}'^{(s)}_l = \begin{cases} \begin{bmatrix} \mathbf{0}_s & \boldsymbol{X}'_l \end{bmatrix} & \text{if } s \geq 0, \\ \begin{bmatrix} \boldsymbol{X}'_l & \mathbf{0}_{-s} \end{bmatrix} & \text{if } s < 0, \end{cases} \tag{D.8}$$

and where $\mathbf{0}_s$ is a $12 \times s$ matrix of zeros. Since the distance measure is based on the Frobenius norm, it obeys the triangle inequality.

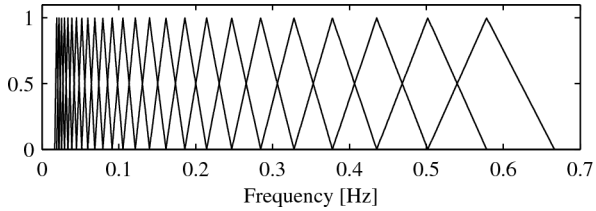


Fig. D.4: Bandwidths of the 25 logarithmically spaced filters.

4 Evaluation

We have evaluated the distance measure by using a nearest neighbor classifier on two different datasets, namely a set of synthesized MIDI files [12] and the covers80 set [13]. Furthermore, the algorithm was evaluated as part of the MIREX 2007 cover song identification task [14].

The basic set of MIDI files consists of 900 MIDI songs that are 30 different melodies of length 180 seconds played with 30 different instruments. To measure the sensitivity to transpositions and variations in tempo, queries that are transposed and lengthened/shortened are used. For each query, the nearest neighbor is found, and the fraction of nearest neighbor songs that share the same melody is counted. In Fig. D.5 the effect of transpositions is shown, and in Fig. D.6 the effect of changing the tempo is shown. It is seen that transposing songs hardly affects performance, and that changing the tempo between a factor 0.7 and 1.4 also does not affect performance too seriously.

The covers80 dataset consists of 80 titles each in two different versions, i.e., a total of 160 songs. The vast majority of the titles have been recorded by two different artists, although a few consist of a live version and a studio version by the same artist. The 160 songs are split into two sets with one version of each song in each set. When evaluating the cover song detection system, the nearest neighbor in the second set to a query from the first set is assumed to be the cover. With this setup, the cover version was found in 38% of the cases. However, as parameters have been tweaked using this dataset, some degree of overtraining is inevitable. In the following, by rank of a cover song we mean rank of the cover when all songs are sorted by their distance to the query. A rank of one means the nearest neighbor to the query song is its cover version, while a rank of e.g. 13 means there are 12 other songs that are considered closer than the real cover by the system. In Fig. D.7, a histogram of the ranks of all the covers is shown. A closer inspection of the data reveals that 66% of the cover songs are within the 10 nearest neighbors. In Table D.1, the songs with the highest ranks are listed. For most of these, the two versions are very different, although a few, such as "Summertime Blues", are actually quite similar. Nevertheless, improving on the

heavy tail is probably not possible without taking lyrics into account.

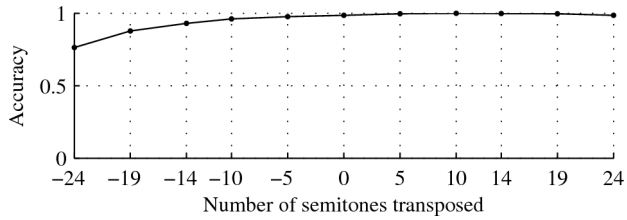


Fig. D.5: Effect of transpositions on melody recognition accuracy.

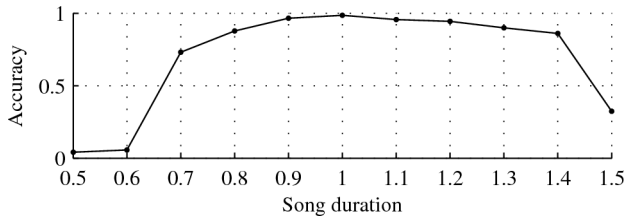


Fig. D.6: Effect of lengthening or shortening a song on melody recognition accuracy. The duration is relative to the original song.

Comparing different music information retrieval algorithms has long been impractical, as copyright issues have prevented the development of standard music collections. The annual MIREX evaluations overcome this problem by having participants submit their algorithms which are then centrally evaluated. This way, distribution of song data is avoided. We submitted the proposed system to the MIREX 2007 audio cover song identification task. The test set is closed and consists of 30 songs each in 11 versions and 670 unrelated songs used as noise. Each of the 330 cover songs are in turn used as query. Results of the evaluation are shown in Table D.2, where it is seen that the proposed system came in fourth. Interestingly, it has almost the exact same performance as the 2006 winner.

5 Conclusion

We have presented a low complexity cover song identification system with moderate storage requirements and with comparable performance to the cover song identification algorithm that performed best at the MIREX 2006 evaluation. Since the proposed distance measure obeys the triangle inequality, it might be useful in large-scale databases. However, further studies are needed to determine

Title	Artists	Rank
My Heart Will Go On	Dion/New Found Glory	74
Summertime Blues	A. Jackson/Beach Boys	71
Yesterday	Beatles/En Vogue	71
Enjoy The Silence	Dep. Mode/T. Amos	60
I Can't Get No Satisfact.	B. Spears/R. Stones	51
Take Me To The River	Al Green/Talking Heads	50
Wish You Were Here	Pink Floyd/Wyclef Jean	50
Street Fighting Man	RATM/R. Stones	48
Tomorrow Never Knows	Beatles/Phil Collins	35
I'm Not In Love	10cc/Tori Amos	33
Red Red Wine	Neil Diamond/UB40	33

Table D.1: Titles of songs with rank > 30.

whether the intrinsic dimensionality of the feature space is too high to utilize this in practice.

Acknowledgements

The authors would like to thank the IMIRSEL team for organizing and running the MIREX evaluations.

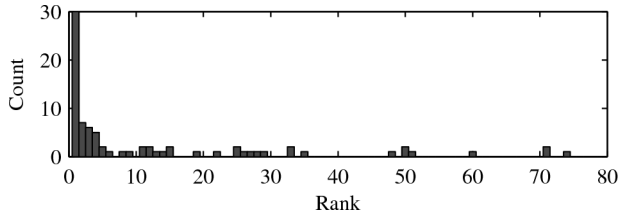


Fig. D.7: Histogram of the cover song ranks.

Rank	Participant	Avg. prec.	Covers in top 10
1	Serrà & Gómez	0.521	1653
2	Ellis & Cotton	0.330	1207
3	Bello, J.	0.267	869
4	*Jensen, Ellis, Christensen & Jensen*	*0.238*	*762*
5	Lee, K. (1)	0.130	425
6	Lee, K. (2)	0.086	291
7	Kim & Perelstein	0.061	190
8	IMIRSEL	0.017	34

Table D.2: MIREX 2007 Audio Cover Song Identification results. In comparison, the 2006 winner [5] identified 761 cover songs in top 10.

References

[1] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proc. IEEE Int. Conf. Multimedia Expo*, Tokyo, Japan, 2001, pp. 745 – 748.

[2] E. Pampalk, "Computational models of music similarity and their application to music information retrieval," Ph.D. dissertation, Vienna University of Technology, Austria, Mar. 2006.

[3] J.-J. Aucouturier, "Ten experiments on the modelling of polyphonic timbre," Ph.D. dissertation, University of Paris 6, France, Jun. 2006.

[4] J. H. Jensen, M. G. Christensen, M. N. Murthi, and S. H. Jensen, "Evaluation of MFCC estimation techniques for music similarity," in *Proc. European Signal Processing Conf.*, Florence, Italy, 2006.

[5] D. P. W. Ellis and G. Poliner, "Identifying cover songs with chroma features and dynamic programming beat tracking," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2007, pp. 1429–1432.

[6] W.-H. Tsai, H.-M. Yu, and H.-M. Wang, "A query-by-example technique for retrieving cover versions of popular songs with similar melodies," in *Proc. Int. Symp. on Music Information Retrieval*, 2005, pp. 183–190.

[7] K. Lee, "Identifying cover songs from audio using harmonic representation," in *Music Information Retrieval Evaluation eXchange*, 2006.

[8] J. P. Bello, "Audio-based cover song retrieval using approximate chord sequences: Testing shifts, gaps, swaps and beats," in *Proc. Int. Symp. on Music Information Retrieval*, 2007, pp. 239–244.

[9] T. Lidy and A. Rauber, "Combined fluctuation features for music genre classification," in *Music Information Retrieval Evaluation eXchange*, 2005.

[10] M. A. Bartsch and G. H. Wakefield, "To catch a chorus: using chroma-based representations for audio thumbnailing," in *Proc. IEEE Workshop on Appl. of Signal Process. to Aud. and Acoust.*, 2001, pp. 15 – 18.

[11] S. Saito, H. Kameoka, T. Nishimoto, and S. Sagayama, "Specmurt analysis of multi-pitch music signals with adaptive estimation of common harmonic structure," in *Proc. Int. Symp. on Music Information Retrieval*, 2005, pp. 84–91.

[12] J. H. Jensen, M. G. Christensen, and S. H. Jensen, "A framework for analysis of music similarity measures," in *Proc. European Signal Processing Conf.*, PoznaŏĬ144, Poland, 2007, pp. 926–930.

[13] D. P. W. Ellis. (2007) The "covers80" cover song data set. [Online]. Available: http://labrosa.ee.columbia.edu/projects/coversongs/covers80/

[14] J. S. Downie, K. West, D. P. W. Ellis, and J. Serrà. (2007, Sep.) MIREX audio 2007 cover song identification. [Online]. Available: http://www.music-ir.org/mirex/2007/index.php/Audio_Cover_Song_Identification

Paper E

A Tempo-insensitive Representation of Rhythmic Patterns

Jesper Højvang Jensen, Mads Græsbøll Christensen, and
Søren Holdt Jensen

This will be published in
Proceedings of the European Signal Processing Conference,
2009.

Abstract

We introduce a representation for rhythmic patterns that is insensitive to minor tempo deviations and that has well-defined behavior for larger changes in tempo. We have combined the representation with an Euclidean distance measure and compared it to other systems in a classification task of ballroom music. Compared to the other systems, the proposed representation shows much better generalization behavior when we limit the training data to songs with different tempi than the query. When both test and training data contain songs with similar tempo, the proposed representation has comparable performance to other systems.

1 Introduction

Together with timbre and melody, rhythm is one of the basic properties of Western music. Nevertheless, it has been somewhat overlooked in the music information retrieval community, perhaps because rhythm is a quite abstract concept that is difficult to describe verbally. A manifestation of this is that in an online music tagging game, Mandel noted that except for the occasional use of the word "beat", hardly any tags were describing rhythm [1]. This suggests that a computational measure of rhythmic distance could supplement a word-based music search engine quite well. An indication that rhythmic similarity has been largely neglected is the audio description contests that were held in conjunction with the International Conference on Music Information Retrieval (ISMIR) in 2004 to compare the performance of different algorithms [2]. Among these evaluations was an automated rhythm classification task, where [3] was the *only* participant. While other tasks such as genre classification were quite popular and have recurred in the Music Information Retrieval Evaluation eXchange (MIREX) as a direct continuation of the ISMIR 2004 evaluation, the rhythm classification task has to date not been repeated. Fortunately, the ballroom music used for the evaluation has been released (see Table E.1 and Figure E.1).

Some of the first systems for rhythm matching were described by Foote et al. [4], who used a self similarity matrix to obtain a beat spectrum that estimates the periodicity of songs at different lags; Paulus and Klapuri [5] who among others use dynamic time warping to match different rhythms; and Tzanetakis and Cook [6] who used an enhanced autocorrelation function of the temporal envelope and a peak picking algorithm to compute a beat histogram as part of a more general genre classification framework. More recent systems include [3, 7–9]. Seyerlehner et al. also use a measure of distance between rhythmic patterns, although with the purpose of tempo estimation [10]. For a review of rhythm description systems, see e.g. [11].

Several authors have observed that tempo is an important aspect of matching

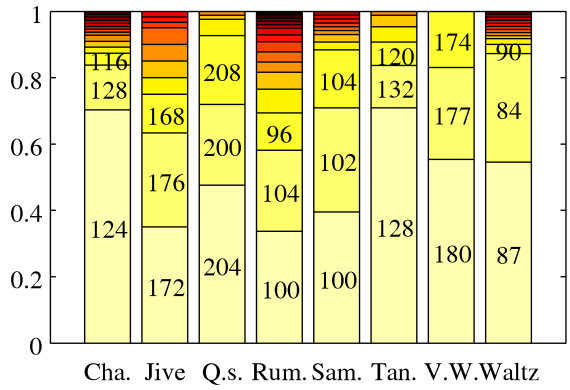


Fig. E.1: Distribution of tempi for the different rhythmic styles in the ballroom dataset. For the three most common number of beat per minutes (BPMs), the value is shown.

songs by rhythm [8, 12, 13]. Using the Ballroom dataset (see Table E.1 and Figure E.1), Gouyon reports a classification accuracy of 82% from the annotated tempi alone, although the accuracy decreases to 53% when using estimated tempi [14]. Peeters reports that combining rhythmic features with the annotated tempi typically increases classification accuracy by around 15% [8]. Seyerlehner et al. have gone even further and have shown that a nearest neighbor classifier that matches the autocorrelation function of the envelope performed on par with state of the art tempo induction systems [10], suggesting that tempo estimation

Table E.1: Distribution of rhythmic styles and training/test split for the music used in the ISMIR 2004 rhythm classification contest. The set consists of 698 clips of ballroom music from `http://ballroomdancers.com/`.

Style	Num. clips	Training	# Test
Cha-cha-cha	111	78	33
Jive	60	42	18
Quickstep	82	57	25
Rumba	98	69	29
Samba	86	60	26
Tango	86	60	26
Viennese Waltz	65	45	20
Waltz	110	77	33

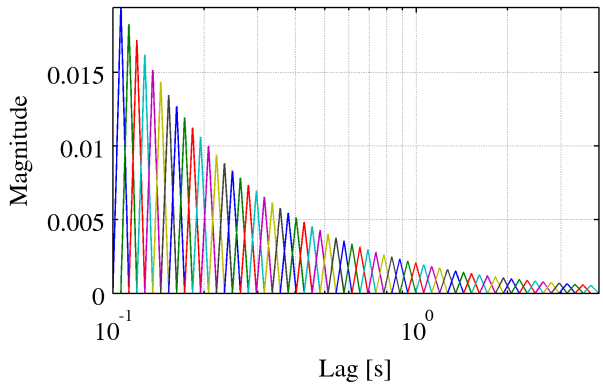


Fig. E.2: The 60 exponentially distributed bands the autocorrelation values are merged into.

can be considered a special case of rhythmic pattern matching. Davies and Plumbley [15] take the opposite approach and use a rhythmic style classifier to improve tempo estimates by letting the prior probabilities of different tempi be a function of the estimated style.

Since rhythm and tempo are so critically linked, we propose a representation of rhythmic patterns that is insensitive to small tempo variations, and where the effect of large variations is very explicit. The representation is based on the melodic distance measures we presented in [16, 17], which were designed to find cover songs, i.e. different renditions of the same song. To make features insensitive to the tempo variations that are inevitable when artists interpret songs differently, we averaged intensities over exponentially spaced bands, which effectively changes a time scaling into a translation. In this paper, we apply the same idea to a measure of rhythmic distance. In Section 2, we describe the proposed representation of rhythmic patterns. In Section 3, we use a nearest neighbor classifier based on the Euclidean distance between the proposed feature to evaluate the performance of the representation on the ballroom dataset. In Section 4, we discuss the results.

2 A tempo-insensitive rhythmic distance measure

Our proposed rhythmic distance measure is inspired by [10], which again is based on [18]. The first steps proceed as in [10, 18]:

1. For each song, resample it to 8 kHz and split it into 32 ms windows with a hop size of 4 ms.

2. For each window, compute the energy in 40 frequency bands distributed according to the mel-scale.

3. For each mel-band, compute the difference along the temporal dimension and truncate negative values to zero to obtain an onset function.

4. Sum the onset functions from all mel-bands into a single, combined onset function. If $P_b(k)$ is the energy of the b'th mel-band in the k'th window, the combined onset function is given by $\sum_b \max(0, P_b(k) - P_b(k-1))$.

5. High-pass filter the combined onset function.

6. Compute the autocorrelation function of the high-pass filtered onset signal up to a lag of 4 seconds.

The autocorrelation function is independent of temporal onset, and it does not change if silence is added to the beginning or end of a song. However, as argued by Peeters [8] it still captures relative phase. While some different rhythmic patterns will share the same autocorrelation function, this is not generally the case. In particular, two rhythmic patterns build from the same durations, (e.g. two $\frac{1}{4}$ notes followed by two $\frac{1}{8}$ notes compared to the sequence $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{8}$) do not in general result in identical autocorrelation functions.

Unlike [10], who smoothes the autocorrelation function on a linear time scale, we use a logarithmic scale. That is, we split the autocorrelation function into the 60 exponentially spaced bands with lags from 0.1 s to 4 s that are shown in Figure E.2. Viewing the energy of the bands on a linear scale corresponds to viewing the autocorrelation function on a logarithmic scale. Changing the tempo of a song would result in a scaling of the autocorrelation function along the x axis by a constant, but on a logarithmic scale, this would be a simple translation. This trick is used in e.g. [19] for fundamental frequency estimation to obtain a representation where the distances between the fundamental frequency and its harmonics are independent of the fundamental frequency. With the exponentially spaced bands, a small change of tempo does not significantly change the distribution of energy between the bands, while larger changes will cause the energy to shift a few bands up or down. We collect the band outputs in a 60-dimensional feature vector $\mathbf{x}$ that has the energy of the n'th band as its n'th component, $(\mathbf{x})_n$. As the final step in the feature extraction process, we normalize the vector to have unit Euclidean norm. In Figure E.3 and E.4, we have shown the proposed feature extracted from the same MIDI file synthesized at three different tempi and from the ballroom dataset, respectively.

With 60 bands, the effective bandwidth of each band extends ±3% from the center frequency. Since a 3% change of tempo is hardly noticeable, in the evaluation we extend the permissible range of tempi by also searching for shifted

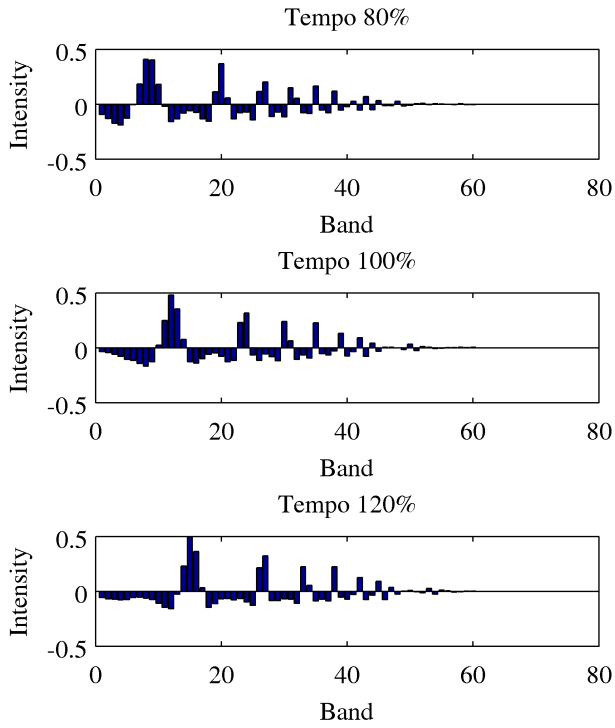


Fig. E.3: The resulting feature vector from a synthesized MIDI file with duration 80%, 100% and 120% of the original length, respectively. Note that the feature vectors are merely shifted versions of each other.

versions of the feature vectors. Specifically, when we search for the nearest neighbor to a song with feature vector $\mathbf{x_m}$, we find the song whose feature vector $\mathbf{x_n}$ is the solution to

$$\arg\min_{\mathbf{n}} \min_{j\in\{-1,0,1\}} \|\mathbf{x_m}^j - \mathbf{x_n}\| \tag{E.1}$$

where $\mathbf{x_m}^j$ is $\mathbf{x_m}$ shifted j steps, i.e.,

$$\mathbf{x_m}^j = \begin{cases} [(\mathbf{x_m})_2 \ (\mathbf{x_m})_3 \ \cdots \ (\mathbf{x_m})_{60} \ 0]^{\mathrm{T}} & \text{for } j = -1, \\ \mathbf{x_m} & \text{for } j = 0, \\ [0 \ (\mathbf{x_m})_1 \ (\mathbf{x_m})_2 \ \cdots \ (\mathbf{x_m})_{59}]^{\mathrm{T}} & \text{for } j = 1. \end{cases} \tag{E.2}$$

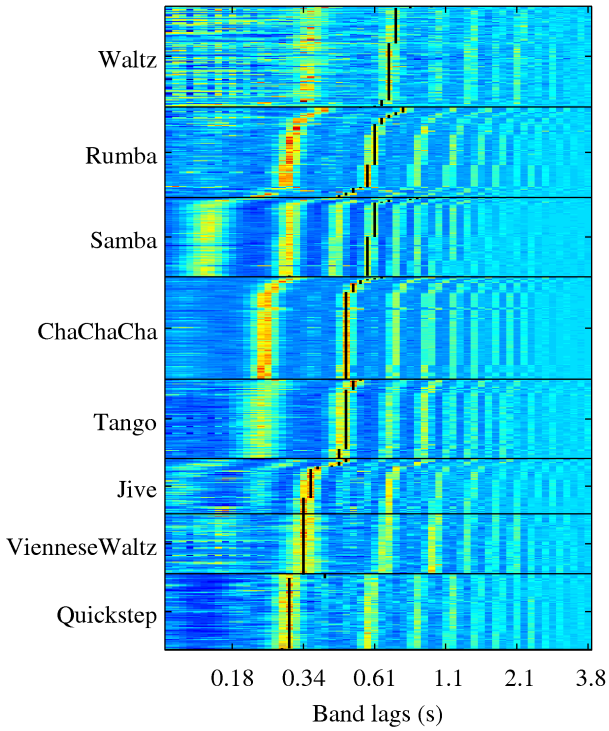


Fig. E.4: Features from the ballroom dataset. Within each style, the features are sorted by the annotated tempo. The band with a lag that corresponds to the annotated tempo (i.e., 120 bpm corresponds to 0.5 s) is indicated by the black, vertical lines. The 60 bands along the x axis are denoted by lag time rather than index.

To obtain something similar with the linear autocorrelation sequence, we would need to resample it to different tempi. However, since the displacement of a peak at lag k is proportional to k, the number of resampled autocorrelation functions must be high to ensure sufficiently high resolution also for large k.

A Matlab implementation of the proposed system is available as part of the Intelligent Sound Processing toolbox[1].

[1] `http://isound.es.aau.dk/`

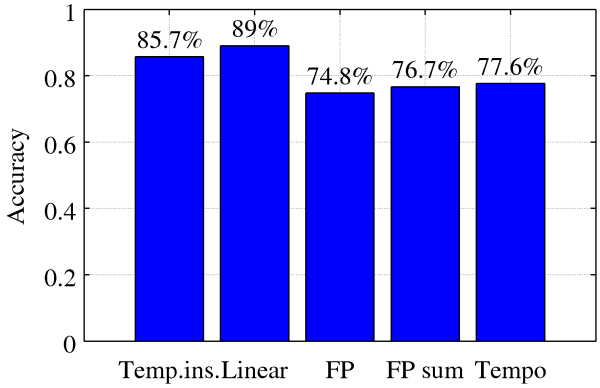


Fig. E.5: Rhythmic style and tempo classification results when allowing the distance measures to match on tempo. From left to right, the distance measures are our proposed tempo insensitive distance measure, the linear version from [10], the Fluctuation Patterns from [20], the modified version of the Fluctuation Patterns from [10], and finally the absolute difference between the songs' ground truth tempi.

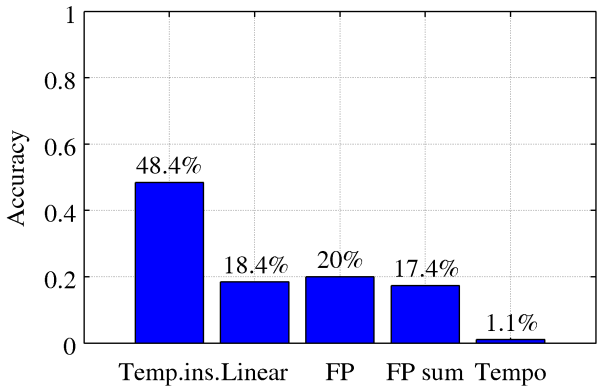


Fig. E.6: Rhythmic style and tempo classification results when ignoring potential nearest neighbors with the same style and similar in tempo to the query.

3 Experiments

Using the ISMIR 2004 ballroom dataset, we have compared the linear autocorrelation as proposed by [10], our proposed logarithmic version, the fluctuation patterns from [20], and the modification to the fluctuation patterns also proposed in [10]. As a reference, we have also used the absolute difference between

the true tempi of songs. We have compared the rhythmic distance measures using two different setups. First, we have used the ballroom dataset as intended by finding the nearest neighbor in the training set to each song in the test set and see how often the rhythmic styles match. These results are shown in Figure E.5. Note that since we use the same partitioning into test and training as in the ISMIR 2004 contest, results are comparable to [2], but although the numbers are similar, the results are not directly comparable to e.g. [8–10] who all use 10-fold cross validation.

To see how much these results depend on tempo, we have repeated the experiment with the difference that when searching for the nearest neighbor to a query song, we reject candidates that have the same rhythmic style and a tempo that is within 4% of the query (we use 4% similarly to [10]). The results when incorporating this constraint are shown in Figure E.6. Test songs with Viennese Waltz had to be ignored when computing the accuracy, since their tempi are all within 5% of each other.

4 Discussion

By constructing a measure of rhythmic distance that is designed to be insensitive to different tempi, we sacrifice a few percentage points of performance in the baseline test in Figure E.5, where the linear autocorrelation function has the highest performance. However, as seen in Figure E.6, if the songs in the training set with the same rhythmic style as the query do not include songs that also share the same tempo, our proposed distance measure significantly outperforms the other distance measures. Due to the good generalization behaviour, we expect our proposed measure to supplement for instance a timbre-based music search engine quite well.

Several aspects of the proposed distance measure are somewhat arbitrary, leaving room for improvement. For example, using other onset functions, e.g. the one used in [21], or using more sophisticated classification algorithms, such as support vector machines, might increase performance.

An interesting aspect of our proposed representation of rhythmic patterns is that by simply shifting the feature vector, it allows searching for slower or faster music with a similar rhythmic structure. This could e.g. be useful if listening to music when exercising, where the push of a button could find similar, faster music that better matches ones pulse.

References

[1] M. I. Mandel and D. P. W. Ellis, "A web-based game for collecting music metadata," in *Proc. Int. Symp. on Music Information Retrieval*, 2007.

[2] P. Cano, E. Gómez, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S. Streich, and N. Wack, "Ismir 2004 audio description contest," Music Technology Group of the Universitat Pompeu Fabra, Tech. Rep., 2006. [Online]. Available: http://www.iua.upf.edu/mtg/files/publications/MTG-TR-2006-02.pdf

[3] T. Lidy and A. Rauber, "Genre-oriented organization of music collections using the somejb system: An analysis of rhythm patterns and other features," in *Proc. of the DELOS Workshop on Multimedia Contents in Digital Libraries*, Chania, Crete, Jun. 2003.

[4] J. Foote, M. Cooper, and U. Nam, "Audio retrieval by rhythmic similarity," in *Proc. Int. Symp. on Music Information Retrieval*, 2002, pp. 265–266.

[5] J. Paulus and A. Klapuri, "Measuring the similarity of rhythmic patterns," in *Proc. Int. Symp. on Music Information Retrieval*, 2002, pp. 150–156.

[6] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Processing*, vol. 10, pp. 293–301, 2002.

[7] S. Dixon, F. Gouyon, and G. Widmer, "Towards characterisation of music via rhythmic patterns," in *Proc. Int. Symp. on Music Information Retrieval*, 2004, pp. 509–516. [Online]. Available: http://ismir2004.ismir.net/proceedings/p093-page-509-paper165.pdf

[8] G. Peeters, "Rhythm classification using spectral rhythm patterns," in *Proc. Int. Symp. on Music Information Retrieval*, 2005, pp. 644–647.

[9] N. Scaringella, "Timbre and rhythmic trap-tandem features for music information retrieval," in *Proc. Int. Symp. on Music Information Retrieval*, 2008, pp. 626–631.

[10] K. Seyerlehner, G. Widmer, and D. Schnitzer, "From rhythm patterns to perceived tempo," in *Proc. Int. Symp. on Music Information Retrieval*, Vienna, Austria, 2008, pp. 519–524.

[11] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer, "A review of automatic rhythm description systems," *Computer Music Journal*, vol. 29, no. 1, pp. 34–54, 2005.

[12] S. Dixon, E. Pampalk, and G. Widmer, "Classification of dance music by periodicity patterns," in *Proc. Int. Symp. on Music Information Retrieval*, 2003. [Online]. Available: http://ismir2003.ismir.net/papers/Dixon.PDF

[13] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer, "Evaluating rhythmic descriptors for musical genre classification," in *Proc. Int. AES Conference*, London, UK, 2004, p. 196ŋ204.

[14] F. Gouyon, "A computational approach to rhythm description — Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing," Ph.D. dissertation, University Pompeu Fabra, Barcelona, Spain, November 2005.

[15] M. E. P. Davies and M. D. Plumbley, "Exploring the effect of rhythmic style classification on automatic tempo estimation," in *Proc. European Signal Processing Conf.*, 2008.

[16] J. H. Jensen, M. G. Christensen, D. P. W. Ellis, and S. H. Jensen, "A tempo-insensitive distance measure for cover song identification based on chroma features," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Las Vegas, USA, 2008, pp. 2209–2212.

[17] J. H. Jensen, M. G. Christensen, and S. H. Jensen, "A chroma-based tempo-insensitive distance measure for cover song identification using the 2D auto-correlation function," in *Music Information Retrieval Evaluation eXchange*, 2008.

[18] D. P. W. Ellis, "Beat tracking with dynamic programming," in *Music Information Retrieval Evaluation eXchange*, 2006.

[19] S. Saito, H. Kameoka, T. Nishimoto, and S. Sagayama, "Specmurt analysis of multi-pitch music signals with adaptive estimation of common harmonic structure," in *Proc. Int. Symp. on Music Information Retrieval*, 2005, pp. 84–91.

[20] E. Pampalk, "A Matlab toolbox to compute music similarity from audio," in *Proc. Int. Symp. on Music Information Retrieval*, 2004, pp. 254–257.

[21] M. Alonso, G. Richard, and B. David, "Accurate tempo estimation based on harmonic + noise decomposition," *EURASIP J. Applied Signal Processing*, vol. 2007, 2007, doi:10.1155/2007/82795.

Paper F

A Framework for Analysis of Music Similarity Measures

Jesper Højvang Jensen, Mads Græsbøll Christensen, and
Søren Holdt Jensen

This paper has been published in
Proceedings of the European Signal Processing Conference,
2007.

The layout has been revised.

Abstract

To analyze specific properties of music similarity measures that the commonly used genre classification evaluation procedure does not reveal, we introduce a MIDI based test framework for music similarity measures. We introduce the framework by example and thus outline an experiment to analyze the dependency of a music similarity measure on the instrumentation of a song compared to the melody, and to analyze its sensitivity to transpositions.

Using the outlined experiment, we analyze music similarity measures from three software packages, namely Marsyas, MA toolbox and Intelligent Sound Processing toolbox. The tested timbral similarity measures perform instrument recognition relatively well, although they are sensitive to transpositions and differences between sound fonts. The beat/rhythm/melody similarity measures are not always able to recognize the same melody played with different instruments.

1 Introduction

As sound compression has matured and storage has become cheap, digital music collections, e.g. in the mp3 format, have grown very large. Navigation in such collections is limited by the metadata, such as title and artist, that is associated with the songs. Appropriate music similarity measures could help navigating in such collections, and could also be used for music recommendation systems in online music stores. By music similarity measure, we mean a quantitative measure of the similarity (or distance) between some musical aspects of two songs. Most music similarity measures are divided into a feature extraction part that extracts features that compactly describe some musical aspects of a song, and a distance measure that computes the distance between songs from the features. Much work has already been done on music similarity and on the related task of genre classification, e.g. [1–9]. Genre classification is often used to evaluate music similarity measures since it simplifies evaluation compared to the numerous user evaluations that are otherwise needed.

While genre classification provides a good first estimate of the performance of a music similarity measure, it does not provide details of its inner workings. For example, a commonly used feature that performs relatively well in genre classification when combined with a classifier, is the mel-frequency cepstral coefficients (MFCCs), e.g. [3–8]. The MFCCs have their origins in speech recognition, where they are used to model the spectral envelope of a single speaker while suppressing the fundamental frequency. This is consistent with the common notion in music similarity of MFCCs as a timbral descriptor. Timbre is defined as "the auditory sensation in terms of which a listener can judge that two sounds with the same loudness and pitch are dissimilar" [10]. Since the temporal envelope and

the spectral envelope play key roles to the perception of timbre [11], one would expect the MFCCs to mainly depend on the instrumentation of a song, and one would expect them to perform genre classification by matching songs with similar instrumentation. It is a tempting conclusion, but there are a number of uncertainties. For instance, in music several notes are often played at once, and it is not obvious how this mixing affects the spectral envelope. Furthermore, it is well-known that MFCCs are not completely independent of the fundamental frequency (e.g. [12]). Unfortunately, the good performance in genre classification does not reveal to which extent the MFCCs reflect the instrumentation, and to which extent they reflect the harmonies and key of a song. In this paper, we take the first steps towards an answer by introducing a test framework based on MIDI synthesis that supplements the genre classification results.

In Section 2, we describe an experiment to evaluate the dependency of a similarity measure on instrumentation compared to melody and an experiment to evaluate the sensitivity of a similarity measure to transpositions. In Section 3 and Section 4, we briefly describe some similarity measures we have evaluated using the experimental setups and present the results, respectively. In Section 5, we discuss the results as well as how the experiments can be modified to analyze other aspects of music similarity measures.

2 Analysis Framework

To analyze how a music similarity measure depends on the instrumentation compared to the notes and how it is affected by transpositions, we use a MIDI file[1] setup. We take a number of MIDI files, manipulate the instrumentation and key using the MATLAB MIDI-toolbox [13], and then use a software synthesizer to generate waveform signals that can be used in a nearest neighbor classifier. Using MIDI files might bias the results, since the synthesized signal will be more homogeneous than recordings of real musicians would be. The advantage is that it allows us to manipulate instrumentation, tempo and melody in a flexible, reproducible way. In what follows, we first introduce an experiment to test the dependency of a musical similarity measure on instrumentation compared to the dependency on the notes, i.e., the melody and harmonies. Second, we introduce an experiment to evaluate the dependency on transpositions of a song, i.e., how shifting all notes of a song a number of semitones affects the similarity measure.

[1] A MIDI file contains information about fundamental frequency, instrumentation and onset/duration of all notes.

2.1 Instrumentation versus notes

The dependency on instrumentation is tested by taking M different MIDI songs and choosing N instruments. We choose songs of different musical styles to ensure that the songs will have very different harmonic and melodic characteristics. For each m, from 1 to M, and for each n, from 1 to N, we do the following:

1. Read MIDI file m.
2. Remove all percussive instruments.
3. Let all notes be played by instrument n.
4. Synthesize a waveform signal.
5. Extract the feature vector $\mathbf{v}_{mn}$ from the waveform signal.

In the following, we will use the term "melody" to denote the instrument and key-independent part of a song, i.e., we will say that all songs created from MIDI file m, which share the exact same melody, harmonies and timing, share the same melody no matter what the instrumentation or key is. After computing features for all $M \times N$ songs, we find the nearest neighbor of $\mathbf{v}_{mn}$ according to the distance $d(\cdot, \cdot)$ associated with the feature:

$$(p, q) = \underset{\substack{l,k \\ (l,k)\neq(m,n)}}{\arg\min}\ d(\mathbf{v}_{mn}, \mathbf{v}_{lk}) \tag{F.1}$$

Let $\mathbf{v}_{mn}$ be a given query, and let the nearest neighbor among the target songs be $\mathbf{v}_{pq}$. If $p = m$, then the nearest neighbor has the same melody as the query. We define the melody classification accuracy by the fraction of the $M \times N$ queries where the nearest neighbor has the same melody. Similarly, we define the instrument classification accuracy as the fraction of queries where the nearest neighbor uses the same instrument.

2.2 Transpositions

A human listener does not consider the same song played in different keys as different songs. Similarly, an instrument playing two different notes is still considered the same instrument. For most similarity measures it is therefore of interest to know how sensitive they are to transpositions. This is what this experiment investigates. It is similar to the previous experiment; the differences being that the tonal range is normalized and the song is transposed. The tonal range is normalized by transposing each individual track of a song (such as bass or melody) by an integer number of octaves, such that the average note being played in a track is as close as possible to the C4 note (middle C on the piano).

The constraint of transposing tracks an integer number of octaves ensures that the harmonic relationships are not changed. As before, let m and n denote melody and instrument number, and let s denote the number of semitones a song is transposed. Features $\mathbf{v}_{mn}^{(s)}$ are computed for different values of s. When evaluating (F.1), a query that has not been transposed is always used, i.e. the minimization is over $d(\mathbf{v}_{mn}^{(0)}, \mathbf{v}_{lk}^{(s)})$. Melody and instrument classification rates are computed for all values of s.

2.3 Implementation of the framework

In genre classification, standard data sets such as the training data from the ISMIR 2004 Genre Classification contest [14] is readily available. However, for our purpose there is no obvious MIDI data set to use. For this reason we created 112 songs of length 30 s using Microsoft Music Producer, a program for automatically creating MIDI files for background music. Each song has a different musical style with different melody, rhythm, tempo, accompaniment and instrumentation. Examples of styles are "50s rock", "Latin" and "Southern rock". From the General MIDI Level 1 Sound Set, all the 112 instruments that neither belong to the percussive instrument family nor are sound effects (see [15]), were selected. Of the 112 instruments and 112 songs, ten random subsets of 30 songs and 30 instruments were chosen. For each subset, the experiments described in Section 2.1 and 2.2 were performed. To synthesize waveform signals from MIDI, the software synthesizer TiMidity++ was used. Two different sound fonts, Fluid R3 and SGM-180 v1.5 GM, were used. Equation (F.1) was evaluated both with query and target synthesized from the same sound fonts, and with query and target synthesized from different sound fonts. All feature extraction routines where given a mono signal sampled at 22 kHz as input.

3 Music similarity measures

In this section, the tested similarity measures are described. Music similarity measures from three different publicly available software packages have been tested: Marsyas [16], the MA toolbox [17], and the Intelligent Sound Processing toolbox (see `http://isound.kom.auc.dk/`). Since not all of the similarity measures incorporate a distance measure between individual songs, some ad hoc distance measures have been introduced. These are also described below.

3.1 Marsyas

From Marsyas v. 0.1, five feature sets are tested. The feature sets are thoroughly described in [3], where they were used with a probabilistic classifier that was

trained on features from an entire genre. For this reason, a distance measure between feature vectors from individual songs does not exist. For all but the *beat* feature, which performed better with ordinary Euclidean distance, we therefore use the weighted Euclidean distance, $d_{\boldsymbol{W}}(\mathbf{u},\mathbf{v}) = (\mathbf{u}-\mathbf{v})^{\mathrm{T}}\boldsymbol{W}(\mathbf{u}-\mathbf{v})$, where $\boldsymbol{W}$ is a diagonal matrix with positive elements. For the timbre features, $\boldsymbol{W}$ was chosen to maximize the difference between the average distance between all vectors and the average distance between vectors from songs with the same instrument, subject to $\|\boldsymbol{W}\|_{\mathrm{F}} = 1$, where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm:

$$\boldsymbol{W} = \underset{\substack{\boldsymbol{W}\\ \|\boldsymbol{W}\|_{\mathrm{F}}=1}}{\arg\max}\left[\frac{1}{M^2N^2}\sum_{i,j}\sum_{k,l} d_{\boldsymbol{W}}(\mathbf{v}_{ij},\mathbf{v}_{kl}) - \frac{1}{M^2N}\sum_{j}\sum_{i,k} d_{\boldsymbol{W}}(\mathbf{v}_{ij},\mathbf{v}_{kj})\right]. \tag{F.2}$$

Before computing $\boldsymbol{W}$, all feature dimensions were normalized to have unit variance. For the pitch feature, the average distance between songs of the same melody was minimized instead. The weights $\boldsymbol{W}$ were computed from one of the 30×30 subsets from the experiment in Section 2.1 where both query and target songs were synthesized with the Fluid R3 sound font. The same weights were used for the transposition experiment. In the following, the five feature sets from Marsyas we have tested are described:

Timbre: Mean and variance of the spectral centroid, roll-off, flux and of the fraction of low-energy frames [3]. Distance measure: Weighted Euclidean.

MFCC: Mean and variance of the first five mel-frequency cepstral coefficients (MFCCs) [3]. Distance measure: Weighted Euclidean.

Timbre + MFCC (T+M): Concatenation of the timbre and MFCC features [3]. Distance measure: Weighted Euclidean.

Beat: Based on a histogram of prominent beats. Consists of the amplitudes and periods of the two first peaks in the histogram, the ratio between these two peaks, and the sum of all peaks [3]. Distance measure: Euclidean.

Pitch: Derived from a histogram of pitches in the signal. Contains among others periods and amplitudes of some of the most prominent peaks on both a full semitone scale and on an octave-independent (modulus 12) scale [3]. Distance measure: Weighted Euclidean.

3.2 MA toolbox

From the MA toolbox [17], five features were tested. The distance measures recommended in [17] are used.

MFCC: MFCCs are estimated in short windows, and a Gaussian mixture model (GMM) is trained to model them. Distance measure: Approximated, symmetrized Kullback-Leibler [7].

Sone: In a number of frequency bands distributed according to the Bark scale, the loudness measured in sone is computed. A GMM is trained on the loudness values. Distance measure: Approximated, symmetrized Kullback-Leibler [7].

Spectrum histogram (SH): A derivative of the raw sone features where the number of times each loudness level has been exceeded in each frequency band is counted [17]. Distance measure: Euclidean.

Periodicity histogram (PH): A description of periodic beats [17]. Distance measure: Euclidean.

Fluctuation pattern (FP): Another approach to describe periodicities in a signal. Distance measure: Euclidean.

3.3 Intelligent Sound Processing toolbox

Two similarity measures from the Intelligent Sound Processing (ISP) toolbox were tested:

MFCC: Similar to the MA toolbox *MFCC*, but with different parameters, such as the number of dimensions. Distance measure: Approximated, symmetrized Kullback-Leibler.

MAR: A multivariate autoregressive model that captures temporal correlation of MFCCs over 1 s segments [18]. A feature vector is produced for every 1 s of audio. Distance measure: For each vector in the query song, the other songs are ranked according to their minimum distance to that vector. The average ranking is then used as the distance measure.

4 Results

The results of the two experiments are plotted in Figures F.1 and F.2. As is seen, some of the results are highly dependent on whether the query features are synthesized from the same sound font as the target features or not. However, the results are largely independent of which of the two sound fonts is used as query and which is used as target. Therefore, only results for Fluid 3 as both query and target, and results for Fluid 3 as query and SGM 180 as target are shown.

When query and target are from the same sound font, the timbral similarity measures perform well. The Marsyas *Timbre+MFCC*, the MA toolbox *MFCC* and *Sone*, and the ISP toolbox *MFCC* all have average instrument classification ratios in the range from 83% to 92%. The Marsyas *MFCC*, MA toolbox *spectrum*

histogram and ISP toolbox *MFCC-MAR* also have relatively good performance, ranging from 75% to 79%. The Marsyas *timbre* performs worst of the timbral features with 55%. However, when query and target are from different sound fonts, the average instrument classification accuracy never exceeds 30%. Since the difference between the same instrument synthesized with different sound fonts is clearly audible, this is understandable, although still undesirable. According to [19], temporal characteristics such as attack transients contribute significantly to human perception of timbre. Timbre similarity measures that better incorporate this short-time temporal development might be less sensitive to the use of different sound fonts.

With respect to melody classification, three similarity measures are noteworthy: Marsyas *beat*, and MA toolbox *periodicity histogram* and *fluctuation pattern* with average classification accuracies of 51%, 78% and 62%, respectively. They are all practically independent of the combination of sound fonts used. The Marsyas *pitch* feature performs surprisingly bad, probably due to the inherently difficult problem of estimating multiple fundamental frequencies. Interestingly, the *fluctuation pattern* from the MA toolbox also performs better than random for instrument classification. Since in this laboratory setup neither the melody, rhythm, accompaniment nor timing changes, it ought to be possible to classify all melodies correctly. We therefore see much room for improvement.

The second experiment shows that all the timbral similarity measures behave similarly when exposed to transpositions. Accuracy is approximately halved when transposing 12 semitones (one octave). When transposing 24 semitones (two octaves), almost no instruments are recognized. An interesting detail is that the behavior of the *MFCC* feature from the MA toolbox is more similar to the *sone* feature from the same toolbox than it is to the *MFCC* feature from the ISP toolbox. The reason might be that the former two use the same statistical model. The features that performed well in melody recognition, namely MA toolbox *periodicity histogram* and *fluctuation pattern* and Marsyas *beat*, are all practically unaffected by transpositions. The Marsyas *pitch* feature is sensitive to transpositions, but since it contains information about the most dominant pitch, this is not surprising.

5 Discussion

From the experiments we observed that the timbral similarity measures did not generalize well to different sound fonts. We therefore hypothesize that timbral similarity measures that also rely on the temporal envelope will better reflect the human sound perception where certain smooth spectral changes, such as adjusting the bass and treble, do not significantly alter the perception of timbre. We also observed that there is room for improvements with melody recognition.

These results could not have been obtained from a genre classification experiment alone. By using MIDI files, we have effectively separated the effect of instrumentation and melody, and a signal modification that would have been difficult or cumbersome to introduce directly in a waveform signal, namely transposition, has been introduced.

Although in this paper we have only tested the sensitivity of similarity measures to transpositions, it would also be relevant to measure the dependency on tempo, combinations of instruments, bandwidth and audio compression. We strongly recommend the use of such tests as a simple, yet insightful supplement to genre classification.

References

[1] J. T. Foote, "Content-based retrieval of music and audio," in *Multimedia Storage and Archiving Systems II, Proc. of SPIE*, 1997, pp. 138–147.

[2] J.-J. Aucouturier and F. Pachet, "Finding songs that sound the same," in *Proc. of IEEE Benelux Workshop on Model based Processing and Coding of Audio*, University of Leuven, Belgium, November 2002.

[3] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Processing*, vol. 10, pp. 293–301, 2002.

[4] A. Flexer, "Statistical evaluation of music information retrieval experiments," Institute of Medical Cybernetics and Artificial Intelligence, Medical University of Vienna, Tech. Rep., 2005.

[5] J.-J. Aucouturier and F. Pachet, "Improving timbre similarity: How high's the sky?" *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.

[6] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proc. IEEE Int. Conf. Multimedia Expo*, Tokyo, Japan, 2001, pp. 745 – 748.

[7] E. Pampalk, "Speeding up music similarity," in *2nd Annual Music Information Retrieval eXchange*, London, Sept. 2005.

[8] M. I. Mandel and D. P. W. Ellis, "Song-level features and support vector machines for music classification," in *Proc. Int. Symp. on Music Information Retrieval*, 2005, pp. 594–599.

[9] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl, "Aggregate features and AdaBoost for music classification," *Machine Learning*, vol. 65, no. 2–3, pp. 473–484, Dec. 2006.

[10] *Acoustical Terminology SI*, New York: American Standards Association Std., Rev. 1-1960, 1960.

[11] B. C. J. Moore, *An introduction to the Psychology of Hearing*, 5th ed. Elsevier Academic Press, 2004.

[12] B. Milner and X. Shao, "Prediction of fundamental frequency and voicing from mel-frequency cepstral coefficients for unconstrained speech reconstruction," *IEEE Trans. Audio, Speech, Language Processing*, vol. 15, no. 1, pp. 24–33, 2007.

[13] T. Eerola and P. Toiviainen, "MIDI toolbox: Matlab tools for music research," University of Jyväskylä: Kopijyvä, Jyväskylä, Finland, Tech. Rep., 2004. [Online]. Available: http://www.jyu.fi/musica/miditoolbox/

[14] ISMIR 2004 audio description contest – genre/artist ID classification and artist similarity. [Online]. Available: http://ismir2004.ismir.net/genre_contest/index.htm

[15] *General MIDI Level 1 Sound Set*, MIDI Manufacturers Association, 2007. [Online]. Available: http://www.midi.org/about-midi/gm/gm1sound.shtml

[16] G. Tzanetakis and P. Cook, "Marsyas: A framework for audio analysis," *Organized Sound*, vol. 4, no. 3, 2000.

[17] E. Pampalk, "A Matlab toolbox to compute music similarity from audio," in *Proc. Int. Symp. on Music Information Retrieval*, 2004, pp. 254–257.

[18] A. Meng, P. Ahrendt, and J. Larsen, "Improving music genre classification by short-time feature integration," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Mar. 2005, pp. 497–500.

[19] T. D. Rossing, F. R. Moore, and P. A. Wheeler, *The Science of Sound*, 3rd ed. Addison-Wesley, 2002.

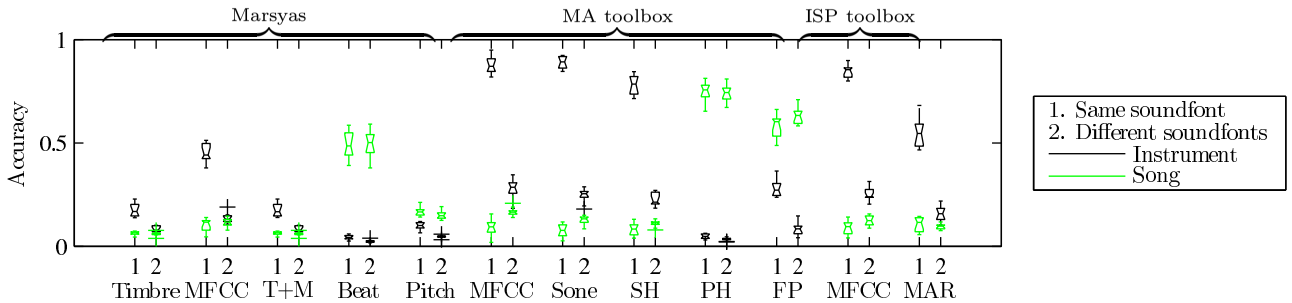


Fig. F.1: Instrument and melody classification accuracies. The number 1 denotes that the same sound font has been used for both query and target, while 2 denote that different sound fonts were used for the query and target. The whiskers on the plot denote the 95% confidence intervals.

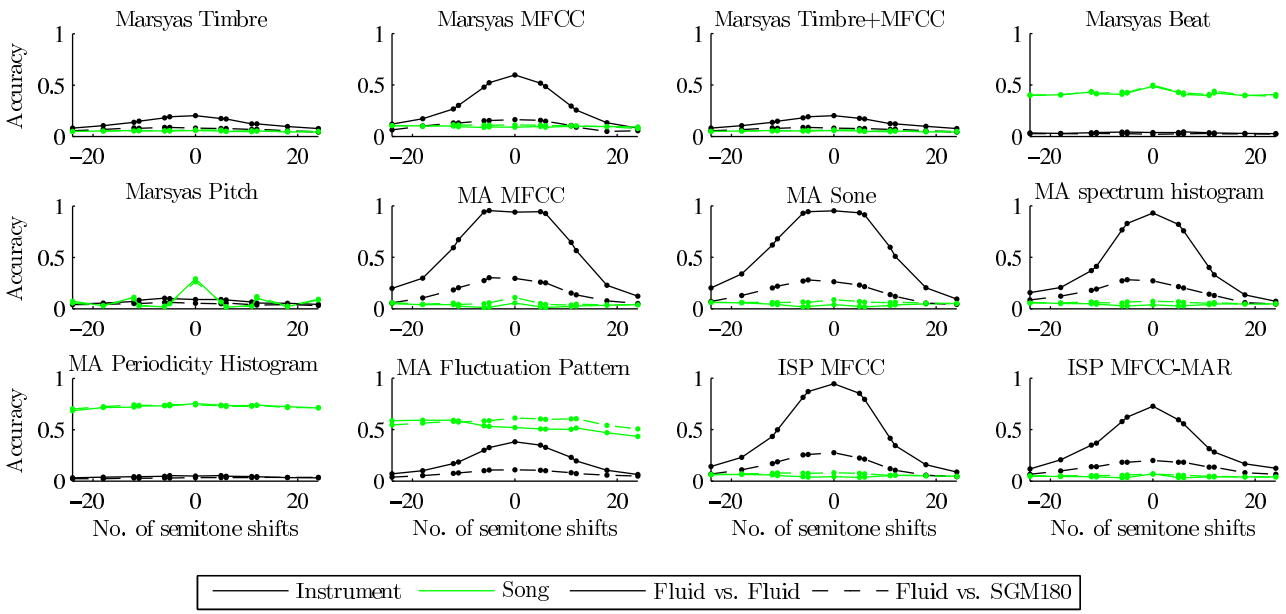


Fig. F.2: Sensitivity of music similarity measures to transpositions.

Paper G

An Amplitude and Covariance Matrix Estimator for Signals in Colored Gaussian Noise

Jesper Højvang Jensen, Mads Græsbøll Christensen, and
Søren Holdt Jensen

This paper will be published in
Proceedings of the European Signal Processing Conference,
2009.

Abstract

We show that by considering the estimation of the amplitudes of sinusoids in colored, Gaussian noise as a joint amplitude and noise covariance matrix estimation problem, we obtain an iterative estimator that has the Capon spectral estimator as a special case. The estimator is also closely related to the amplitude and phase estimator (APES). In experiments, the proposed joint estimator in most cases outperforms Capon and APES.

1 Introduction

The estimation of the amplitudes of sinusoids in colored, Gaussian noise has a long history with applications such as radar imaging and audio coding (see e.g. [1] and the references therein). The simplest approach is to ignore the coloring of the noise and use methods such as least squares that assume white noise. A refinement is e.g. the Capon spectral estimator that uses the signal covariance matrix as an estimate of the noise covariance [2] (see also [3, 4]). An evolution of this is the amplitude and phase estimator (APES), which uses a cheap amplitude estimate to obtain a refined noise covariance estimate [4, 5]. A related approach for audio signals can be found in [6]. The single-sinusoid APES algorithm was originally derived as an approximation to the exact joint maximum likelihood amplitude and noise covariance matrix estimator [5], similar to what we propose here for multiple sinusoids. However, when the multiple sinusoids APES algorithm was derived in [4], the noise covariance matrix was estimated prior to the amplitudes, not jointly.

In this paper, we do not consider the estimation of the noise covariance matrix and the sinusoid amplitudes as two separate tasks, but rather estimate them jointly. The resulting estimator is indeed closely related to both the Capon amplitude estimator and APES. While the joint estimator is computationally more demanding than the former two, it has the advantage that it avoids ad hoc noise covariance estimates.

In Section 2, we derive the joint noise covariance and sinusoid amplitude estimator, in Section 3 we evaluate it, and in Section 5 we conclude on the proposed approach.

2 Maximum Likelihood Parameter Estimation

We are concerned with the following complex-valued signal model:

$$x(n) = e(n) + \sum_{l=1}^{L} \alpha_l \exp(i\omega_l n) \text{ for } n \in \{0, 1, \ldots, N-1\}, \qquad \text{(G.1)}$$

where $x(n)$ is the observed signal, ω_l are the known frequencies of the complex sinusoids, α_l are the unknown, complex amplitudes, and $e(n)$ is complex, colored, zero-mean Gaussian noise. The assumption of zero-mean noise is without loss of generality, since a non-zero mean is equivalent to an additional sinusoid with a frequency of zero. We define the vectors

$$\boldsymbol{x}(n) = \begin{bmatrix} x(n) & \cdots & x(n+M-1) \end{bmatrix}^{\mathrm{T}}, \tag{G.2}$$

$$\boldsymbol{e}(n) = \begin{bmatrix} e(n) & \cdots & e(n+M-1) \end{bmatrix}^{\mathrm{T}}, \tag{G.3}$$

$$\boldsymbol{s}_l(n) = \begin{bmatrix} \exp(i\omega_l n) & \cdots & \exp(i\omega_l(n+M-1)) \end{bmatrix}^{\mathrm{T}}, \tag{G.4}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \cdots & \alpha_L \end{bmatrix}^{\mathrm{T}}, \tag{G.5}$$

and the matrix

$$\boldsymbol{S}(n) = \begin{bmatrix} \boldsymbol{s}_1(n) & \cdots & \boldsymbol{s}_L(n) \end{bmatrix}. \tag{G.6}$$

Letting $G = N - M + 1$ be the number of observed vectors, (G.1) is equivalent to

$$\boldsymbol{x}(n) = \boldsymbol{S}(n)\boldsymbol{\alpha} + \boldsymbol{e}(n) \text{ for } n \in \{0, 1, \ldots, G-1\}. \tag{G.7}$$

We will assume that the noise vectors $\boldsymbol{e}(n)$ are independent and Gaussian with $M \times M$ covariance matrix $\boldsymbol{Q} = \mathrm{E}[\boldsymbol{e}(n)\boldsymbol{e}(n)^{\mathrm{H}}]$. In this case, the maximum likelihood estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{Q}$, denoted by $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{Q}}$, are given by

$$[\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{Q}}] = \arg\max_{\boldsymbol{\alpha}, \boldsymbol{Q}} J(\boldsymbol{\alpha}, \boldsymbol{Q}) \tag{G.8}$$

where

$$J(\boldsymbol{\alpha}, \boldsymbol{Q}) = \frac{1}{G} \sum_{n=0}^{G-1} \log p(x(n)) \tag{G.9}$$

$$= \frac{1}{G} \sum_{n=0}^{G-1} \log \left[\frac{1}{\pi^M |\boldsymbol{Q}|} \exp\left(-\boldsymbol{e}(n)^{\mathrm{H}} \boldsymbol{Q}^{-1} \boldsymbol{e}(n)\right) \right], \tag{G.10}$$

with $\boldsymbol{e}(n) = \boldsymbol{x}(n) - \boldsymbol{S}(n)\boldsymbol{\alpha}$.

We find $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{Q}}$ in a two-step process. We first find the maximum likelihood estimate of $\hat{\boldsymbol{Q}}$ given the amplitudes, i.e.,

$$\hat{\boldsymbol{Q}}(\boldsymbol{\alpha}) = \arg\max_{\boldsymbol{Q}} J(\boldsymbol{\alpha}, \boldsymbol{Q}). \tag{G.11}$$

Next, inserting $\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})$ in J, we find the $\boldsymbol{\alpha}$ that maximizes J with $\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})$ inserted for $\boldsymbol{Q}$:

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}, \hat{\boldsymbol{Q}}(\boldsymbol{\alpha})). \tag{G.12}$$

If all values exist and are unique, $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{Q}}(\hat{\boldsymbol{\alpha}})$ found in this way are identical to $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{Q}}$ found by directly maximizing (G.8) (this can be proven by assuming the opposite and show that this leads to contradictions). Using this procedure, we show that maximizing the log-likelihood of the observed data is equivalent to minimizing the determinant of the estimated noise covariance matrix, before we derive an iterative estimator that finds this minimum.

It is well-known that the maximum likelihood estimate of the noise covariance matrix given the amplitudes is given by [1]

$$\hat{\boldsymbol{Q}}(\boldsymbol{\alpha}) = \frac{1}{G}\sum_{n=0}^{G-1} \boldsymbol{e}(n)\boldsymbol{e}(n)^{\mathrm{H}}, \tag{G.13}$$

still with $\boldsymbol{e}(n) = \boldsymbol{x}(n) - \boldsymbol{S}(n)\boldsymbol{\alpha}$.

To see that maximizing the log-likelihood is equivalent to minimizing the determinant of the estimated noise covariance matrix (as shown by e.g. [7] and [5]), we first insert (G.13) in (G.10):

$$J(\boldsymbol{\alpha}, \hat{\boldsymbol{Q}}(\boldsymbol{\alpha})) = \frac{1}{G}\sum_{n=0}^{G-1} \log\left[\frac{1}{\pi^M |\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})|} \mathrm{e}^{-\boldsymbol{e}(n)^{\mathrm{H}}\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{-1}\boldsymbol{e}(n)}\right] \tag{G.14}$$

$$= -M\log\pi - \log|\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})| - \frac{1}{G}\sum_{n=0}^{G-1} \boldsymbol{e}(n)^{\mathrm{H}}\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{-1}\boldsymbol{e}(n). \tag{G.15}$$

Traditionally in maximum likelihood amplitude estimation, the first two terms of (G.15) are considered constant, while the last term is maximized. However, when $\hat{\boldsymbol{Q}}$ is given by (G.13) instead of being independent of the estimated amplitudes, we can show that the last term vanishes, leaving only the second term to be optimized. Inserting (G.13) in the last term, we obtain

$$\begin{aligned}\frac{1}{G}\sum_{n=0}^{G-1} \boldsymbol{e}(n)^{\mathrm{H}}\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{-1}\boldsymbol{e}(n) &= \mathrm{tr}\Big[\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{-1}\frac{1}{G}\sum_{n=0}^{G-1} \boldsymbol{e}(n)\boldsymbol{e}(n)^{\mathrm{H}}\Big] \\ &= \mathrm{tr}\big[\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{-1}\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})\big] = M.\end{aligned} \tag{G.16}$$

In the above, tr[·] denotes the trace operator. Thus, $J(\boldsymbol{\alpha}, \hat{\boldsymbol{Q}}(\boldsymbol{\alpha}))$ is given by

$$J(\boldsymbol{\alpha}, \hat{\boldsymbol{Q}}(\boldsymbol{\alpha})) = -M\log\pi - \log|\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})| - M. \tag{G.17}$$

The maximum likelihood estimates $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{Q}}$ are therefore simply given by

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}, \hat{\boldsymbol{Q}}(\boldsymbol{\alpha})) \tag{G.18}$$

$$= \arg\min_{\boldsymbol{\alpha}} \log|\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})| \tag{G.19}$$

and

$$\hat{\boldsymbol{Q}} = \hat{\boldsymbol{Q}}(\hat{\boldsymbol{\alpha}}). \tag{G.20}$$

For $\boldsymbol{S}(n)$ being a single sinusoid, a closed form expression for $\hat{\alpha}$ was derived in [5]. To compute $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{Q}}$ in the general case, we find the derivative of $\log|\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})|$ with respect to $\boldsymbol{\alpha}$ and find where it equals zero. First, we use the chain rule to see that

$$\frac{d\log|\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})|}{d\alpha_l} = \mathrm{tr}\left(\frac{\partial \log|\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})|}{\partial \hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{\mathrm{T}}}\frac{\partial \hat{\boldsymbol{Q}}(\boldsymbol{\alpha})}{\partial \alpha_l}\right). \tag{G.21}$$

Using the relation $\frac{\partial \log|\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})|}{\partial \hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{\mathrm{T}}} = \hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{-1}$ (see [8, 9]), we first compute the partial derivative with respect to a single amplitude:

$$\begin{aligned}
\frac{d\log|\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})|}{d\alpha_l} &= \mathrm{tr}\left(\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{-1}\frac{\partial \hat{\boldsymbol{Q}}(\boldsymbol{\alpha})}{\partial \alpha_l}\right) && \text{(G.22)}\\
&= \mathrm{tr}\left(\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{-1}\frac{\partial}{\partial \alpha_l}\frac{1}{G}\sum_{n=0}^{G-1}(\boldsymbol{x}(n)-\boldsymbol{S}(n)\boldsymbol{\alpha})(\boldsymbol{x}(n)-\boldsymbol{S}(n)\boldsymbol{\alpha})^{\mathrm{H}}\right) && \text{(G.23)}\\
&= -\frac{1}{G}\,\mathrm{tr}\left(\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{-1}\sum_{n=0}^{G-1}\boldsymbol{s}_l(n)(\boldsymbol{x}(n)-\boldsymbol{S}(n)\boldsymbol{\alpha})^{\mathrm{H}}\right) && \text{(G.24)}\\
&= -\frac{1}{G}\,\mathrm{tr}\left(\sum_{n=0}^{G-1}(\boldsymbol{x}(n)-\boldsymbol{S}(n)\boldsymbol{\alpha})^{\mathrm{H}}\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{-1}\boldsymbol{s}_l(n)\right) && \text{(G.25)}\\
&= -\frac{1}{G}\sum_{n=0}^{G-1}(\boldsymbol{x}(n)-\boldsymbol{S}(n)\boldsymbol{\alpha})^{\mathrm{H}}\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{-1}\boldsymbol{s}_l(n). && \text{(G.26)}
\end{aligned}$$

From this we see that the derivative with respect to $\boldsymbol{\alpha}$ is given by

$$\frac{d\log|\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})|}{d\boldsymbol{\alpha}^{\mathrm{T}}} = \frac{1}{G}\sum_{n=0}^{G-1}(\boldsymbol{x}(n)-\boldsymbol{S}(n)\boldsymbol{\alpha})^{\mathrm{H}}\hat{\boldsymbol{Q}}(\boldsymbol{\alpha})^{-1}\boldsymbol{S}(n). \tag{G.27}$$

Setting this equal to zero, we obtain

$$\hat{\boldsymbol{\alpha}} = \left[\sum_{n=0}^{G-1}\boldsymbol{S}(n)^{\mathrm{H}}\hat{\boldsymbol{Q}}(\hat{\boldsymbol{\alpha}})^{-1}\boldsymbol{S}(n)\right]^{-1}\left[\sum_{n=0}^{G-1}\boldsymbol{S}(n)^{\mathrm{H}}\hat{\boldsymbol{Q}}(\hat{\boldsymbol{\alpha}})^{-1}\boldsymbol{x}(n)\right]. \tag{G.28}$$

However, this is not a closed form solution, since the $\hat{\boldsymbol{Q}}(\hat{\boldsymbol{\alpha}})$ term also depends on $\hat{\boldsymbol{\alpha}}$. Nevertheless, we can choose an initial guess for $\hat{\boldsymbol{\alpha}}$, e.g. $\hat{\boldsymbol{\alpha}}_0 = \boldsymbol{0}$, and

cyclically compute refined covariance matrix and amplitude estimates:

$$\hat{\boldsymbol{Q}}(\hat{\boldsymbol{\alpha}}_k) = \frac{1}{G}\sum_{n=0}^{G-1}(\boldsymbol{x}(n) - \boldsymbol{S}(n)\hat{\boldsymbol{\alpha}}_k)(\boldsymbol{x}(n) - \boldsymbol{S}(n)\hat{\boldsymbol{\alpha}}_k)^{\mathrm{H}} \tag{G.29}$$

and

$$\hat{\boldsymbol{\alpha}}_{k+1} = \left[\sum_{n=0}^{G-1}\boldsymbol{S}(n)^{\mathrm{H}}\hat{\boldsymbol{Q}}(\hat{\boldsymbol{\alpha}}_k)^{-1}\boldsymbol{S}(n)\right]^{-1}\left[\sum_{n=0}^{G-1}\boldsymbol{S}(n)^{\mathrm{H}}\hat{\boldsymbol{Q}}(\hat{\boldsymbol{\alpha}}_k)^{-1}\boldsymbol{x}(n)\right]. \tag{G.30}$$

Note that the maximum likelihood amplitude estimator for sinusoids in additive Gaussian noise with known covariance matrix is identical to (G.30) with $\hat{\boldsymbol{Q}}(\boldsymbol{\alpha}_k)$ replaced by the true covariance matrix. This is surprising, since the latter maximizes the last term of (G.15), while the joint estimator maximizes the second term. Since (G.29) is the maximum likelihood estimate of the covariance matrix given the amplitudes, we cyclically compute the maximum likelihood estimate of one parameter using the current estimate of the other parameter. This guarantees that the log-likelihood of each refined estimate never decreases, whereby convergence is guaranteed. It is an open question, though, whether convergence is always to the global optimum or if it is only to local optima. In practice, we have observed that when using the zero vector as the initial estimate of $\boldsymbol{\alpha}$, the iterations converge in very few iterations.

Using the zero vector as the initial estimate, the first estimate of $\boldsymbol{Q}$ computed from (G.29) becomes

$$\hat{\boldsymbol{Q}}(\boldsymbol{0}) = \frac{1}{G}\sum_{n=0}^{G-1}\boldsymbol{x}(n)\boldsymbol{x}(n)^{\mathrm{H}}, \tag{G.31}$$

and the first non-trivial amplitude estimate becomes

$$\hat{\boldsymbol{\alpha}}_1 = \left[\sum_{n=0}^{G-1}\boldsymbol{S}(n)^{\mathrm{H}}\hat{\boldsymbol{Q}}(\boldsymbol{0})^{-1}\boldsymbol{S}(n)\right]^{-1}\left[\sum_{n=0}^{G-1}\boldsymbol{S}(n)^{\mathrm{H}}\hat{\boldsymbol{Q}}(\boldsymbol{0})^{-1}\boldsymbol{x}(n)\right], \tag{G.32}$$

which is nothing but the Capon amplitude estimator. The Capon estimator is thus the special case of the proposed joint estimator where we use the zero vector as the initial amplitude estimate and stop after a single iteration.

The APES algorithm uses a cheap estimate of the sinusoids' amplitudes to obtain a refined noise covariance estimate. For the multiple sinusoid version in [4], the refined noise covariance matrix is given by

$$\hat{\boldsymbol{Q}} = \frac{1}{G}\sum_{n=0}^{G-1}\boldsymbol{x}(n)\boldsymbol{x}(n)^{\mathrm{H}} - \sum_{l=1}^{L}\boldsymbol{g}_l\boldsymbol{g}_l^{\mathrm{H}}, \tag{G.33}$$

where

$$\boldsymbol{g}_l = \frac{1}{G}\sum_{n=0}^{G-1} \boldsymbol{x}(n)\exp(-i\omega_l n). \tag{G.34}$$

This has some resemblance to the second iteration of the proposed estimator, where the first amplitude estimate is used to obtain a refined noise covariance estimate. However, since no further iterations are performed with the APES algorithm, it does not converge to the maximum likelihood estimate.

Note that in showing that maximizing the log-likelihood is equivalent to minimizing the estimated noise covariance matrix, we only use that the same error signal estimate is used for calculating the noise covariance estimate and for computing the log-likelihood. Hence, this property holds for all signals in additive Gaussian noise, whether it is a linear or a nonlinear combination of deterministic signals, and it also holds if we use the forward-backward estimate for both the covariance matrix and the log-likelihood. In deriving the iterative estimator in (G.29) and (G.30), we restrict ourselves to linear combinations of signals, although they do not need to be sinusoids. Furthermore, since the function we end up optimizing in (G.19) is the determinant of the noise covariance matrix, we can analyze its properties using asymptotic eigenvalue properties. In particular, if the covariance matrix is Toeplitz and obeys certain regularity properties, a special case of Szegő's theorem asymptotically relates the log determinant to the logarithm of the Fourier spectrum of the autocorrelation function [10].

3 Evaluation

We have tested the proposed estimator in the same setup as used in [4], where a sum of three complex sinusoids are buried in autoregressive Gaussian noise. The observed signal is given by

$$x(n) = e(n) + \sum_{l=1}^{3} \alpha_l \exp(i2\pi f_l n) \tag{G.35}$$

with $\alpha_1 = \exp(i\pi/4)$, $\alpha_2 = \exp(i\pi/3)$, $\alpha_3 = \exp(i\pi/4)$, and $f_1 = 0.1$, $f_2 = 0.11$, and $f_3 = 0.3$. The colored noise $e(n)$ is given by

$$e(n) = 0.99e(n-1) + v(n), \tag{G.36}$$

where $v(n)$ is white, Gaussian noise. The observation length N is 32, and the dimension of the covariance matrix, M, is 8. The mean square error for the first and third sinusoid are shown in Figure G.1 and G.2, respectively, for different signal to noise ratios. The mean square error of the first sinusoid, located at frequency 0.1 with a neighboring sinusoid at 0.11, shows how well the different

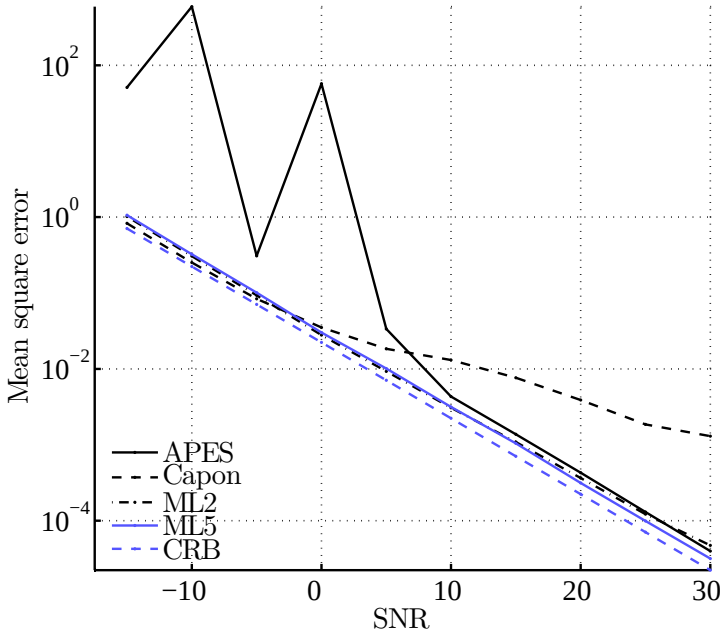


Fig. G.1: The mean square estimation error of α_1 in (G.35), located at $f_1 = 0.1$ with f_2 interfering at 0.11. ML2 and ML5 are the proposed estimator with two and five iterations, respectively, and CRB is the Cramér-Rao lower bound.

estimators handle spectral leakage from sinusoids at neighboring frequencies. The mean square errors of the third sinusoid, located at 0.3, are indicative of performance when no other sinusoids are close. Consequently, the mean square errors in Figure G.2 are much lower than in Figure G.1. The mean square errors are averaged over 1000 realizations. We have compared APES and Capon to the proposed estimator with two and five iterations of (G.29) and (G.30).

In Figure G.1, we see that in the case of an interfering neighboring sinusoid, the proposed estimator has uniformly good performance at all SNR and is consistently close to the Cramér-Rao lower bound. At low SNR, additional iterations decrease performance slightly, and the Capon amplitude estimator (which corresponds to a single iteration) has best performance. At higher SNR, the proposed estimator with five iterations performs best. In Figure G.2, the estimation errors are much lower due to the less dominant interference. APES and the proposed estimator with five iterations seem to have nearly identical performance close to the Cramér-Rao bound, while two iterations and the Capon estimator sees a performance decrease at high SNR.

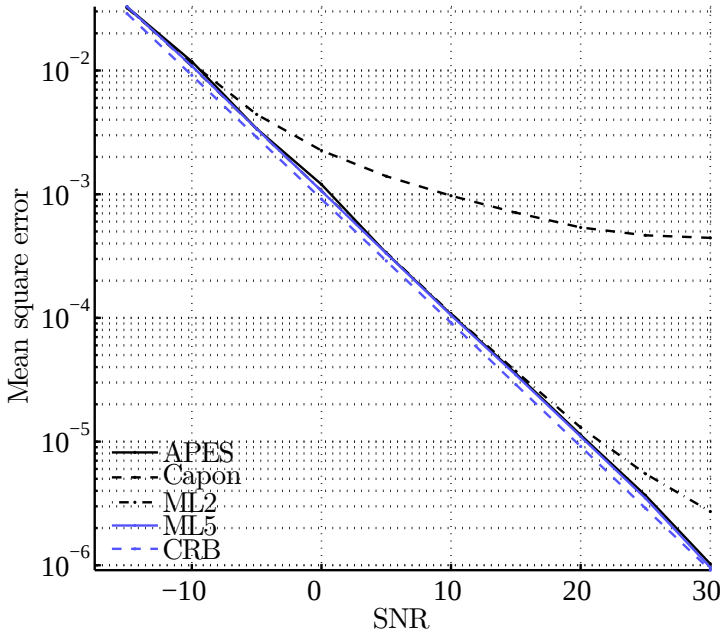


Fig. G.2: The mean square estimation error of α_3 in (G.35), located at $f_3 = 0.3$ with the closest interfering sinusoid located at 0.11.

4 Conclusion

We have shown that maximizing the log-likelihood of a signal in colored, Gaussian noise is equivalent to minimizing the determinant of the estimated noise covariance matrix, and we have derived an iterative algorithm to find the optimum for a sum of sinusoids with unknown amplitudes. The derived algorithm has the Capon amplitude estimator as a special case, and experimentally, the new estimator shows consistently good performance. The proposed estimator has interesting theoretical implications, since it demonstrates that sinusoidal amplitude estimation in colored noise can elegantly be cast as a joint amplitude and noise covariance matrix estimation problem, instead of using ad hoc noise covariance estimates, and because it allows the use of asymptotic determinant properties such as Szegő's theorem for the analysis of maximum likelihood estimators. The latter may also be useful for deriving computationally cheap, asymptotically efficient estimators.

References

[1] P. Stoica and R. Moses, *Spectral Analysis of Signals*. Prentice Hall, 2005.

[2] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, no. 8, pp. 1408–1418, Aug. 1969.

[3] M. A. Lagunas, M. E. Santamaria, A. Gasull, and A. Moreno, "Maximum likelihood filters in spectral estimation problems," *Elsevier Signal Processing*, vol. 10(1), pp. 19–34, 1986.

[4] P. Stoica, H. Li, and J. Li, "Amplitude estimation of sinusoidal signals: survey, new results, and an application," *IEEE Trans. Signal Processing*, vol. 48, no. 2, pp. 338–352, Feb. 2000.

[5] J. Li and P. Stoica, "An adaptive filtering approach to spectral estimation and sar imaging," *IEEE Trans. Signal Processing*, vol. 44, no. 6, pp. 1469–1484, June 1996.

[6] M. G. Christensen and S. H. Jensen, "Variable order harmonic sinusoidal parameter estimation for speech and audio signals," in *Proc. Fortieth Asilomar Conference on Signals, Systems and Computers ACSSC '06*, 2006, pp. 1126–1130.

[7] G. Bienvenu and L. Kopp, "Optimality of high resolution array processing using the eigensystem approach," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 31, no. 5, pp. 1235–1248, 1983.

[8] A. Hjorungnes and D. Gesbert, "Complex-valued matrix differentiation: Techniques and key results," *IEEE Trans. Signal Processing*, vol. 55, no. 6, pp. 2740–2746, June 2007.

[9] T. P. Minka, "Old and new matrix algebra useful for statistics," Tech. Rep., Dec. 2000. [Online]. Available: http://research.microsoft.com/~minka/papers/matrix/minka-matrix.pdf

[10] R. M. Gray, "Toeplitz and circulant matrices: A review," *Foundations and Trends in Communications and Information Theory*, vol. 2, pp. 155–239, 2006.

Paper H

On Optimal Filter Designs for Fundamental Frequency Estimation

Mads Græsbøll Christensen, Jesper Højvang Jensen,
Andreas Jakobsson, and Søren Holdt Jensen

This paper has been published in
IEEE Signal Processing Letters,
vol. 15, pp. 745-748, 2008.

The layout has been revised.

Abstract

Recently, we proposed using Capon's minimum variance principle to find the fundamental frequency of a periodic waveform. The resulting estimator is formed such that it maximises the output power of a bank of filters. We present an alternative optimal single filter design, and then proceed to quantify the similarities and differences between the estimators using asymptotic analysis and Monte Carlo simulations. Our analysis shows that the single filter can be expressed in terms of the optimal filterbank, and that the methods are asymptotically equivalent, but generally different for finite length signals.

1 Introduction

Bandlimited periodic waveforms can be decomposed into a finite set of sinusoids having frequencies that are integer multiples of a so-called fundamental frequency. Much research has been devoted to the problem of finding the fundamental frequency, and rightfully so. It is an important problem in many applications in, for example, speech and audio processing, and the problem has become no less relevant with the many interesting new applications in music information retrieval. The fundamental estimation problem can be mathematically defined as follows: a signal consisting of a set of harmonically related sinusoids related by the fundamental frequency ω_0 is corrupted by an additive white complex circularly symmetric Gaussian noise, $w(n)$, having variance σ^2, for $n = 0, \ldots, N-1$, i.e.,

$$x(n) = \sum_{l=1}^{L} \alpha_l e^{j\omega_0 l n} + w(n), \tag{H.1}$$

where $\alpha_l = A_l e^{j\psi_l}$, with $A_l > 0$ and ψ_l being the amplitude and the phase of the lth harmonic, respectively. The problem of interest is to estimate the fundamental frequency ω_0 from a set of N measured samples $x(n)$. Some representative examples of the various types of methods that are commonly used for fundamental frequency estimation are: linear prediction [1], correlation [2], subspace methods [3], frequency fitting [4], maximum likelihood (e.g., [5]), Bayesian estimation [6], and comb filtering [7]. The basic idea of the comb filtering approach is that when the teeth of the comb filter coincide with the frequencies of the individual harmonics, the output power of the filter is maximized. This idea is conceptually related to our approach derived in [5]; however, here we design optimal signal-adaptive filters reminiscent of beamformers for coherent signals, e.g. [8], for the estimation of the fundamental frequency. In particular, we consider two fundamental frequency estimators based on the well-known minimum variance principle [9]. The two estimators are based on different filter design

formulations with one being based on a bank of filters and the other on only a single filter. The first of these estimators was recently proposed [5], while the second one is novel. The estimators are compared and the asymptotic properties of the estimators are analyzed and their finite length performance is investigated and compared in Monte Carlo simulations. For simplicity, we will here consider only the single pitch estimation problem but the presented methods can easily be applied to multi pitch estimation as well (see [5]).

The remainder of this paper is organized as follows. First, we introduce the two filter designs and the associated estimators in Section 2. Then, we analyze and compare the estimators and their asymptotic properties in Section 3. Their finite length performance is investigated in Section 4, before we conclude on our work in Section 5.

2 Optimal Filter Designs

2.1 Filterbank Approach

We begin by introducing some useful notation, definitions and review the fundamental frequency estimator proposed in [5]. First, we construct a vector from M consecutive samples of the observed signal, i.e., $\mathbf{x}(n) = [\, x(n)\; x(n-1)\; \cdots\; x(n-M+1)\,]^T$ with $M \leq N$ and with $(\cdot)^T$ denoting the transpose. Next, we introduce the output signal $y_l(n)$ of the lth filter having coefficients $h_l(n)$ as $y_l(n) = \sum_{m=0}^{M-1} h_l(m)x(n-m) = \mathbf{h}_l^H\mathbf{x}(n)$, with $(\cdot)^H$ denoting the Hermitian transpose and $\mathbf{h}_l = [\, h_l(0)\; \ldots\; h_l(M-1)\,]^H$. Introducing the expected value $\mathrm{E}\{\cdot\}$ and defining the covariance matrix as $\mathbf{R} = \mathrm{E}\left\{\mathbf{x}(n)\mathbf{x}^H(n)\right\}$, the output power of the lth filter can be written as

$$\mathrm{E}\left\{|y_l(n)|^2\right\} = \mathrm{E}\left\{\mathbf{h}_l^H\mathbf{x}(n)\mathbf{x}^H(n)\mathbf{h}_l\right\} = \mathbf{h}_l^H\mathbf{R}\mathbf{h}_l. \tag{H.2}$$

The total output power of all the filters is $\sum_{l=1}^{L}\mathrm{E}\left\{|y_l(n)|^2\right\} = \sum_{l=1}^{L}\mathbf{h}_l^H\mathbf{R}\mathbf{h}_l$. Defining a matrix $\mathbf{H}$ consisting of the filters $\mathbf{h}_l$ as $\mathbf{H} = [\, \mathbf{h}_1\; \cdots\; \mathbf{h}_L\,]$, we can write the total output power as a sum of the power of the subband signals, i.e., $\sum_{l=1}^{L}\mathrm{E}\left\{|y_l(n)|^2\right\} = \mathrm{Tr}\left[\mathbf{H}^H\mathbf{R}\mathbf{H}\right]$. The filter design problem can now be stated. We seek to find a set of filters that pass power undistorted at specific frequencies, here the harmonic frequencies, while minimizing the power at all other frequencies. This problem can be formulated mathematically as the optimization problem:

$$\min_{\mathbf{H}} \mathrm{Tr}\left[\mathbf{H}^H\mathbf{R}\mathbf{H}\right] \quad \text{s.t.} \quad \mathbf{H}^H\mathbf{Z} = \mathbf{I}, \tag{H.3}$$

where $\mathbf{I}$ is the $L \times L$ identity matrix. Furthermore, the matrix $\mathbf{Z} \in \mathbb{C}^{M\times L}$ has a Vandermonde structure and is constructed from L complex sinusoidal vectors

as

$$\mathbf{Z} = [\ \mathbf{z}(\omega_0)\ \cdots\ \mathbf{z}(\omega_0 L)\ ], \tag{H.4}$$

with $\mathbf{z}(\omega) = [\ 1\ e^{-j\omega}\ \cdots\ e^{-j\omega(M-1)}\ ]^T$. Or in words, the matrix contains the harmonically related complex sinusoids. The filter bank matrix $\mathbf{H}$ solving (H.3) is given by (see, e.g., [10])

$$\mathbf{H} = \mathbf{R}^{-1}\mathbf{Z}\left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}. \tag{H.5}$$

This data and frequency dependent filter bank can then be used to estimate the fundamental frequencies by maximizing the power of the filter's output, yielding

$$\hat{\omega}_0 = \arg\max_{\omega_0} \mathrm{Tr}\left[\left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}\right], \tag{H.6}$$

which depends only on the covariance matrix and the Vandermonde matrix constructed for different candidate fundamental frequencies.

2.2 An Alternative Approach

We proceed to examine an alternative formulation of the filter design problem and state its optimal solution. Suppose that we wish to design a single filter, $\mathbf{h}$, that passes the signal undistorted at the harmonic frequencies and suppresses everything else. This filter design problem can be stated as

$$\min_{\mathbf{h}} \mathbf{h}^H\mathbf{R}\mathbf{h} \quad \text{s.t.} \quad \mathbf{h}^H\mathbf{z}(\omega_0 l) = 1, \quad \text{for} \quad l = 1, \ldots, L. \tag{H.7}$$

It is worth stressing that the single filter in (H.7) is designed subject to L constraints, whereas in (H.3) the filter bank is formed using a matrix constraint. Clearly, these two formulations are related; we will return to this relation in detail in the following section. Introducing the Lagrange multipliers $\boldsymbol{\lambda} = [\ \lambda_1\ \ldots \lambda_L\ ]$, the Lagrangian dual function associated with the problem stated above can be written as

$$L(\mathbf{h}, \boldsymbol{\lambda}) = \mathbf{h}^H\mathbf{R}\mathbf{h} - \left(\mathbf{h}^H\mathbf{Z} - \mathbf{1}^T\right)\boldsymbol{\lambda} \tag{H.8}$$

with $\mathbf{1} = [\ 1\ \ldots\ 1\ ]^T$. Taking the derivative with respect to the unknown filter impulse response, $\mathbf{h}$ and the Lagrange multipliers, we get

$$\nabla L(\mathbf{h}, \boldsymbol{\lambda}) = \begin{bmatrix} \mathbf{R} & -\mathbf{Z} \\ -\mathbf{Z}^H & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{h} \\ \boldsymbol{\lambda} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}. \tag{H.9}$$

By setting this expression equal to zero, i.e., $\nabla L(\mathbf{h}, \boldsymbol{\lambda}) = \mathbf{0}$, and solving for the unknowns, we obtain the optimal Lagrange multipliers for which the equality constraints are satisfied as $\boldsymbol{\lambda} = \left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}\mathbf{1}$ and the optimal filter as

$\mathbf{h} = \mathbf{R}^{-1}\mathbf{Z}\boldsymbol{\lambda}$. By combining the last two expressions, we get the optimal filter expressed in terms of the covariance matrix and the Vandermonde matrix $\mathbf{Z}$, i.e.,

$$\mathbf{h} = \mathbf{R}^{-1}\mathbf{Z}\left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}\mathbf{1}. \tag{H.10}$$

The output power of this filter can then be expressed as

$$\mathbf{h}^H\mathbf{R}\mathbf{h} = \mathbf{1}^H\left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}\mathbf{1}, \tag{H.11}$$

which, as for the first design, depends only on the inverse of $\mathbf{R}$ and the Vandermonde matrix $\mathbf{Z}$. By maximizing the output power, we readily obtain an estimate of the fundamental frequency as

$$\hat{\omega}_0 = \arg\max_{\omega_0} \mathbf{1}^H\left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}\mathbf{1}. \tag{H.12}$$

3 Analysis

We will now relate the two filter design methods and the associated estimators in (H.6) and (H.12). It is perhaps not clear whether the two methods are identical or if there are some subtle differences. On one hand, the optimization problem in (H.3) allows for more degrees of freedom, since L filters of length M are designed while (H.7) involves only a single filter. On the other hand, the former design is based on L^2 constraints as opposed to the latter approach only involving L. Comparing the optimal filters in (H.5) and (H.10), we observe that the latter can be written in terms of the former as

$$\mathbf{h} = \mathbf{R}^{-1}\mathbf{Z}\left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}\mathbf{1} = \mathbf{H}\mathbf{1} = \sum_{l=1}^{L}\mathbf{h}_l, \tag{H.13}$$

so, clearly, the two methods are related. Using this to rewrite the output power in (H.11), we get

$$\mathbf{h}^H\mathbf{R}\mathbf{h} = \left(\sum_{l=1}^{L}\mathbf{h}_l^H\right)\mathbf{R}\left(\sum_{m=1}^{L}\mathbf{h}_m\right) \tag{H.14}$$

as opposed to $\mathrm{Tr}\left[\mathbf{H}^H\mathbf{R}\mathbf{H}\right] = \sum_{l=1}^{L}\mathbf{h}_l^H\mathbf{R}\mathbf{h}_l$ for the filterbank approach. It can be seen that the single-filter approach includes the cross-terms $\mathbf{h}_l^H\mathbf{R}\mathbf{h}_m$ for $l \neq m$, while these do not appear in the filterbank approach. From this it follows that the cost functions are generally different, i.e.,

$$\mathbf{1}^H\left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}\mathbf{1} \neq \mathrm{Tr}\left[\left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}\right] \tag{H.15}$$

$$\mathbf{h}^H\mathbf{R}\mathbf{h} \neq \mathrm{Tr}\left[\mathbf{H}^H\mathbf{R}\mathbf{H}\right]. \tag{H.16}$$

This means that the two filters will result in different output powers and thus possibly different estimates. Next, we will analyze the asymptotic properties of the cost function

$$\lim_{M\to\infty} M\mathbf{1}^H \left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}\mathbf{1}. \tag{H.17}$$

In doing so we will make use of the following result (see, e.g., [11])

$$\lim_{M\to\infty}(\mathbf{AB}) = \left(\lim_{M\to\infty}\mathbf{A}\right)\left(\lim_{M\to\infty}\mathbf{B}\right) \tag{H.18}$$

where it is assumed that the limits $\lim_{M\to\infty}\mathbf{A}$ and $\lim_{M\to\infty}\mathbf{B}$ exist for the individual elements of $\mathbf{A}$ and $\mathbf{B}$. Using (H.18) to rewrite the limit of $\mathbf{I} = \mathbf{AA}^{-1}$, we get

$$\lim_{M\to\infty}\mathbf{I} = \left(\lim_{M\to\infty}\mathbf{A}\right)\left(\lim_{M\to\infty}\mathbf{A}^{-1}\right). \tag{H.19}$$

Next, suppose we have an analytic expression for the limit of $\lim_{M\to\infty}\mathbf{A}$, say, $\bar{\mathbf{A}}$, then we have $\mathbf{I} = \bar{\mathbf{A}}\left(\lim_{M\to\infty}\mathbf{A}^{-1}\right)$ from which we conclude that $(\lim_{M\to\infty}\mathbf{A}^{-1}) = \bar{\mathbf{A}}^{-1}$ and thus

$$\left(\lim_{M\to\infty}\mathbf{A}^{-1}\right) = \left(\lim_{M\to\infty}\mathbf{A}\right)^{-1}. \tag{H.20}$$

Applying (H.18) and (H.20) to the cost function in (H.23), yields

$$\lim_{M\to\infty} M\mathbf{1}^H \left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}\mathbf{1} = \mathbf{1}^H\left(\lim_{M\to\infty}\left(\frac{1}{M}\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)\right)^{-1}\mathbf{1}. \tag{H.21}$$

We are now left with the problem of determining the limit $\lim_{M\to\infty}\frac{1}{M}\left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)$. In doing so, we will make use of the asymptotic equivalence of Toeplitz and circulant matrices. For a given Toeplitz matrix, here $\mathbf{R}$, we can construct an asymptotically equivalent circulant $M\times M$ matrix $\mathbf{C}$, under certain conditions, in the sense that [12] $\lim_{M\to\infty}\frac{1}{\sqrt{M}}\|\mathbf{C}-\mathbf{R}\|_F = 0$, where $\|\cdot\|_F$ is the Frobenius norm and the limit is taken over the dimensions of $\mathbf{C}$ and $\mathbf{R}$. A circulant matrix $\mathbf{C}$ has the eigenvalue decomposition $\mathbf{C} = \mathbf{Q\Gamma Q}^H$ where $\mathbf{Q}$ is the Fourier matrix. Thus, the complex sinusoids in $\mathbf{Z}$ are asymptotically eigenvectors of $\mathbf{R}$. This allows us to determine the limit as (see [12, 13])

$$\lim_{M\to\infty}\frac{1}{M}\left(\mathbf{Z}^H\mathbf{RZ}\right) = \operatorname{diag}\left(\left[\ \Phi(\omega_0)\ \cdots\ \Phi(\omega_0 L)\ \right]\right) \tag{H.22}$$

with $\Phi(\omega)$ being the power spectral density of $x(n)$. Similarly, an expression for the inverse of $\mathbf{R}$ can be obtained as $\mathbf{C}^{-1} = \mathbf{Q\Gamma}^{-1}\mathbf{Q}^H$ (again, see [12] for details). We now arrive at the following (see also [13] and [14]):

$$\lim_{M\to\infty}\frac{1}{M}\left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right) = \operatorname{diag}\left(\left[\ \Phi^{-1}(\omega_0)\ \cdots\ \Phi^{-1}(\omega_0 L)\ \right]\right). \tag{H.23}$$

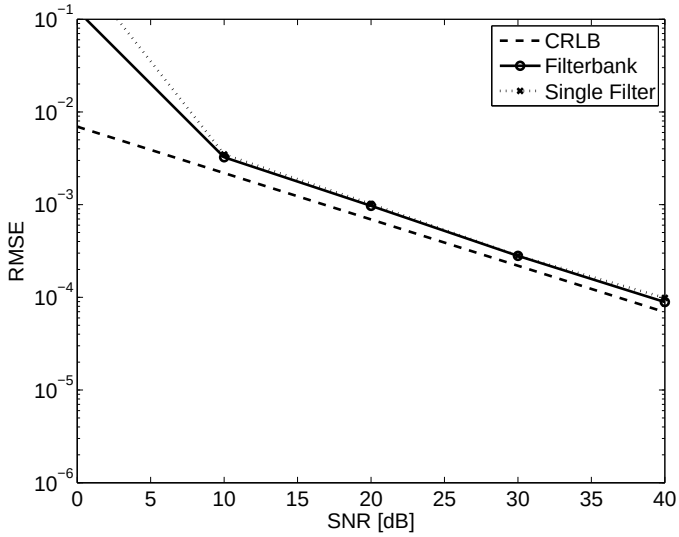


Fig. H.1: RMSE as a function of the SNR for $N = 50$.

Asymptotically, (H.12) can therefore be written as

$$\lim_{M\to\infty} M\mathbf{1}^H \left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}\mathbf{1} = \sum_{l=1}^{L}\Phi(\omega_0 l), \tag{H.24}$$

which is simply the sum over the power spectral density evaluated at the harmonic frequencies. Similar derivations for the filterbank formulation yield

$$\lim_{M\to\infty} M\,\mathrm{Tr}\left[\left(\mathbf{Z}^H\mathbf{R}^{-1}\mathbf{Z}\right)^{-1}\right] = \sum_{l=1}^{L}\Phi(\omega_0 l). \tag{H.25}$$

which is the same as (H.24). Note that for a finite M the above expression still involves only the diagonal terms (due to the trace), only the diagonal terms are not the power spectral density $\Phi(\omega)$ evaluated in certain points. From the above derivations, we conclude that the two cost functions are different for finite M and may yield different estimates, but are asymptotically equivalent.

4 Experimental Results

The question remains to be answered whether there are any important differences for finite length covariance matrices and filters, and we will now seek to answer

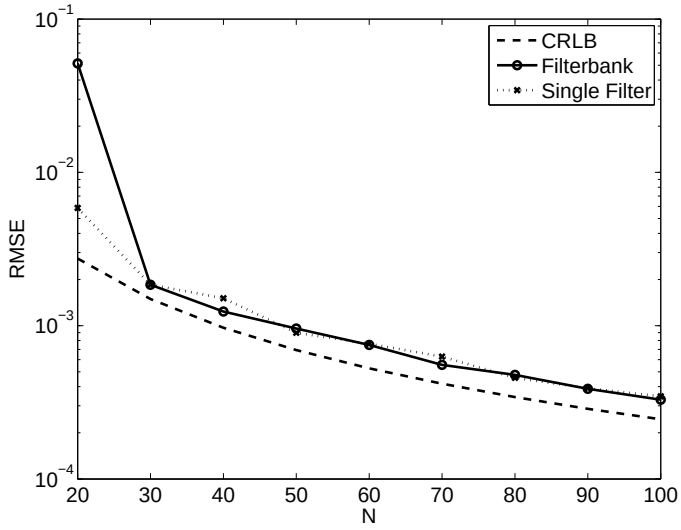


Fig. H.2: RMSE as a function the number of samples N for $SNR = 20$ dB.

that question with some experiments, specifically using Monte Carlo simulations with synthetic signals generated according to (H.1). For each realization, the sample covariance matrix is estimated as $\widehat{\mathbf{R}} = \frac{1}{N-M+1} \sum_{n=0}^{N-M} \mathbf{x}(n)\mathbf{x}^H(n)$ which is used in place of the true covariance matrix. Since both methods require that $\widehat{\mathbf{R}}$ is invertible, we obviously have that $M < \frac{N}{2}$ and in practice we use $M = \left\lfloor \frac{2}{5}N \right\rfloor$, a value that has been determined empirically to yield good results. First, we will investigate the accuracy of the obtained fundamental frequency estimates measured in terms of the root mean square estimation error (RMSE). We do this for $\omega_0 = 0.6364$ with $L = 3$, unit amplitudes, and random phases drawn from a uniform probability density function. In Figure H.1, the RMSE is plotted for $N = 50$ as a function of the signal-to-noise ratio (SNR) (as defined in [3] for the problem in (H.1)). The RMSE was estimated using 200 different realizations. Similarly, the RMSE is shown as a function of the number of samples, N, in Figure H.2 for an SNR of 20 dB., again for 200 realizations. In both figures, the Cramér-Rao lower bound (CRLB), as derived in [3], is also shown. Both figures suggest that, all things considered, there is very little difference in terms of accuracy for the estimated parameters, with both estimators performing well. The methods seem to have different thresholding behaviour, though. We note that our simulations also show that the methods perform similarly as a function of ω_0, but in the interest of brevity, this figure has not been included herein. Next, we will measure the differences of the estimated output powers. We measure

this using the following power ratio (PR):

$$PR = 10\log_{10}\frac{\mathrm{E}\left\{\mathrm{Tr}\left[\left(\mathbf{Z}^H\widehat{\mathbf{R}}^{-1}\mathbf{Z}\right)^{-1}\right]\right\}}{\mathrm{E}\left\{\mathbf{1}^H\left(\mathbf{Z}^H\widehat{\mathbf{R}}^{-1}\mathbf{Z}\right)^{-1}\mathbf{1}\right\}} \quad [\mathrm{dB}], \tag{H.26}$$

which is positive if the output power of the filterbank exceeds that of the single filter and vice versa. It should be noted that the expectation is taken over the realizations of the sample covariance matrix $\widehat{\mathbf{R}}$. The power ratio (averaged over 1000 realizations) is shown in Figure H.3 as a function of the filter length M for an SNR of 10 dB. The filter length is related to the number of samples as $M = \left\lfloor \frac{2}{5}N \right\rfloor$. The fundamental frequency was drawn from a uniform distribution in the interval $[0.1571; 0.3142]$ with $L = 5$ in this experiment to avoid any biases due to special cases. The true fundamental frequency was used in obtaining the optimal filters. In Figure H.4, the same is plotted for $N = 100$, this time as a function of the number of harmonics L with all other conditions being the same as before. Interestingly, both Figures H.3 and H.4 paint a rather clear picture: for low filter lengths and high number of harmonics, the single filter design method actually leads to a better estimate of the signal power while for high filter orders and few harmonics, the methods tend to perform identically. This suggests that the single filter design method is preferable.

5 Conclusion

We have presented two different optimal filter designs that can be used for finding high-resolution estimates of the fundamental frequency of periodic signals. The two designs differ in that one is based on the design of a filterbank while the other is based on a single filter. We have shown that the optimal single filter can in fact be obtained from the optimal filters of the filterbank and that the methods are in fact different for finite lengths, but are asymptotically equivalent. Experiments indicate that the single filter leads to superior results in terms of estimating the output power.

References

[1] K. W. Chan and H. C. So, "Accurate frequency estimation for real harmonic sinusoids," *IEEE Signal Processing Lett.*, vol. 11(7), pp. 609–612, Jul. 2004.

[2] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am.*, vol. 111(4), pp. 1917–1930, Apr. 2002.

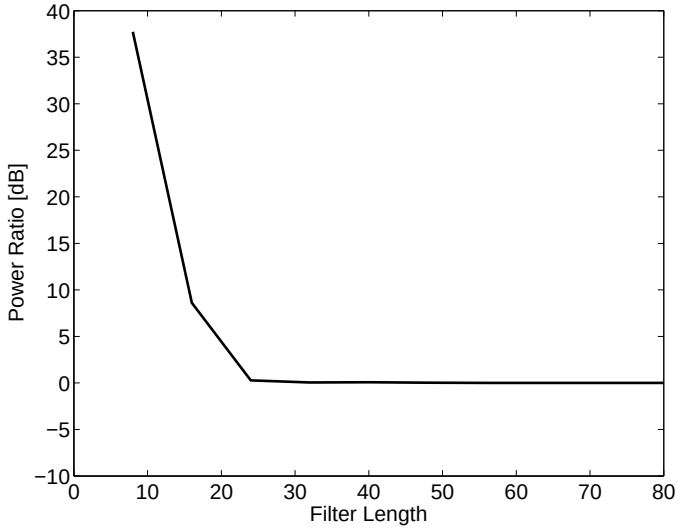


Fig. H.3: Power ratio in dB as a function of the filter length M.

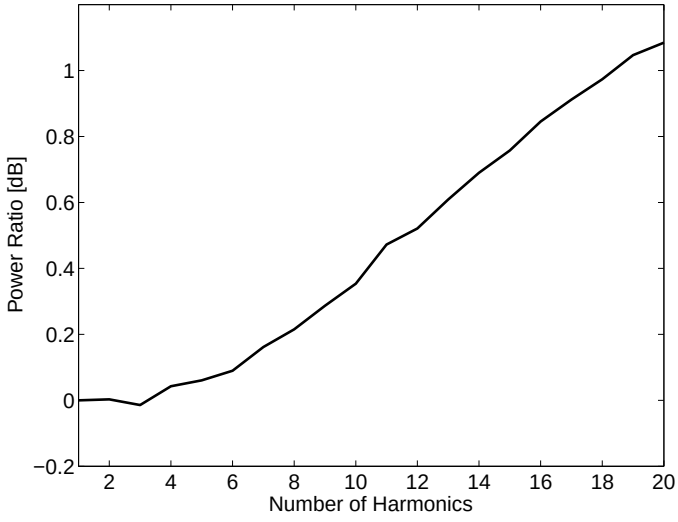


Fig. H.4: Power ratio in dB as a function of the number of harmonics L.

[3] M. G. Christensen, A. Jakobsson and S. H. Jensen, "Joint high-resolution fundamental frequency and order estimation," *IEEE Trans. on Audio,*

Speech and Language Processing, vol. 15(5), pp. 1635–1644, Jul. 2007.

[4] H. Li, P. Stoica, and J. Li, "Computationally efficient parameter estimation for harmonic sinusoidal signals," *Signal Processing*, vol. 80, pp. 1937–1944, 2000.

[5] M. G. Christensen, P. Stoica, A. Jakobsson and S. H. Jensen, "Multi-pitch estimation," *Elsevier Signal Processing*, vol. 88(4), pp. 972–983, Apr. 2008.

[6] S. Godsill and M. Davy, "Bayesian harmonic models for musical pitch estimation and analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, 2002, pp. 1769–1772.

[7] A. Nehorai and B. Porat, "Adaptive comb filtering for harmonic signal enhancement," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34(5), pp. 1124–1138, Oct. 1986.

[8] L. Zhang, H. C. So, H. Ping, and G. Liao, "Effective Beamformer for Coherent Signal Reception," *IEE Electronic Letters*, vol. 39(13), pp. 949–951, Jun. 2003.

[9] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57(8), pp. 1408–1418, 1969.

[10] P. Stoica and R. Moses, *Spectral Analysis of Signals.* Pearson Prentice Hall, 2005.

[11] T. M. Apostol, *Mathematical Analysis*, 2nd ed. Addison-Wesley, 1974.

[12] R. M. Gray, "Toeplitz and circulant matrices: A review," *Foundations and Trends in Communications and Information Theory*, vol. 2(3), pp. 155–239, 2006.

[13] E. J. Hannan and B. Wahlberg, "Convergence rates for inverse toeplitz matrix forms," *J. Multivariate Analysis*, vol. 31, pp. 127–135, 1989.

[14] P. Stoica, H. Li, and J. Li, "Amplitude estimation of sinusoidal signals: Survey, new results and an application," *IEEE Trans. Signal Processing*, vol. 48(2), pp. 338–352, Feb. 2000.