Thematically-Driven Guitar Music Retrieval

Daniel Filipe Pombo da Silva Baptista
daniel.silva.baptista@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

May 2016

Abstract

In the last decade Query-by-example methods have received a huge interest by the MIREX community. Audio matching is commonly used for a query-by-example model. However current audio matching techniques ignore sequential information. In this work, we describe the main approaches for audio matching and propose the adaptation of some of those techniques, dynamic time warping and locally weighted bag-of-words to incorporate sequential information. With this adaptation our goal is to provide a query-by-example with better audio matching and the ability of dealing with underlying relations between the features that are known to exist in the musical domain.

Keywords: Music Information Retrieval, Query-by-Example, Audio Matching, Locally Weighted Bag-of-words, Dynamic Time Warping, Harmony Progression Analyzer, MIREX

1. Introduction

Large music collections are easily accessible due to rapid evolving technology, providing new opportunities for research using these collections to discover trends and patterns in music [5]. With music information retrieval, it is possible to develop a strategy to compute similarity between segments of songs, in order to provide new services, such as, detecting copyright violations or query by example.

Music teachers, students, and even hobbyists, sometimes need more music material to practice their technical skills, instead of repeating the same pattern/exercise over and over. Students often end up in a situation where they are practicing a technique in a song due to a specific section and have the need for more songs with section/sections similar to that one. The time spent searching is sometimes longer than the time available for practice since it is not easy finding songs with similar sections in a large database.

Query-by-example methods have been target of interest by the MIREX community in the last decade. These methods differ from audio classification as its key issue is how to model each audio segment rather than each audio category [11]. The issue of audio segment modeling is not to be taken lightly [5]. In the mid-to-low audio similarity specificity range, the user seeks specific musical content of the query audio but not necessarily the same audio content. These are among the most challenging problems for audio similarity-based MIR systems with less specific retrieval tasks still mostly unsolved.

In thematically-driven retrieval, or fragment-level retrieval scenarios, the query consists of a short fragment of an audio recording. The goal is to find all fragments of a given music collection that are related to the query even though entire songs are returned as matches. Typically, such fragments may cover only a few seconds of audio content or may correspond to a theme, or a musical part of a recording.

We can further specify this retrieval being thematically-driven audio matching where the goal is to retrieve all audio fragments that musically correspond to a query fragment from all audio documents contained in a given database. In this case, one explicitly allows semantically motivated variations since they typically occur in different arrangements and performances of a piece of music. These variations include significant non-linear global and local differences in tempo, articulation, and phrasing as well as differences in executing note groups such as grace notes, trills, or arpeggios [6].

The problem we face is the creation of a query-by-example system that thematically retrieves musical pieces given a musical fragment. There are some difficulties inherent to this problem for instance how similar are two songs? There are objective and subjective similarity measures, however for experimental purposes objective measures are preferred. Another difficulty is in the feature extraction process, this can interfere with how we model a song and how we perform audio matching. Another problem is the music genre, such as classical, pop, rock, among others. Finally the key issue lies in the

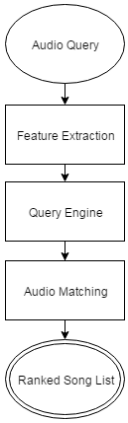

Figure 1: Typical Stages for Query-by-Example.

method we use to perform our audio modeling.

Given a musical fragment as input, produce as output a ranked list of thematically related musical pieces based on tempo, articulation and phrasing. Our system will be based on the locally weighted bag-of-words [14, 13] to easily identify thematic trends across songs. Popular music will be used in this work for evaluation purposes given its wide corpus availability.

This thesis is divided into three main logical parts. The first part (chapter 2) is related to the state-of-the-art which describes how audio matching is done as of the writing of this thesis. We start by reviewing a continuous probability distribution model, Gaussian Mixture Models in audio similarity retrieval, we analyze bag-of-audio-words approaches similar to the bag-of-words model used in text processing, and we review applications of topic models to audio retrieval, such as Hierarchical Dirichlet Process and Latent Dirichlet Allocation.

The second part (chapters 3 and 4) describe the work and subsequent achieved results. It starts with a case study regarding feature extraction and audio matching and respective experiments/results, then the construction of the query-by-example system and obtained results with the datasets.

The third part (chapter 5) is made of conclusions as well as the definition of future work to be done.

2. Background

Query-by-Example systems share a common three stage architecture as shown in figure 1.

From the audio query, systems start by using feature extraction module followed by a query engine, followed by an audio matching stage. Table 1 describes the main responsibility of each stage. The following subsections will explain each of the mentioned steps in greater detail.

For audio matching to work we must first obtain features that represent the audio signal with a certain degree of robustness. Examples of those

Stage	Responsibility
Feature Extraction	Audio signal characteristics
Query Engine	Robust feature modeling
Audio Matching	Distance comparison and ranked sort

Table 1: Typical Query by Example Architecture Description.

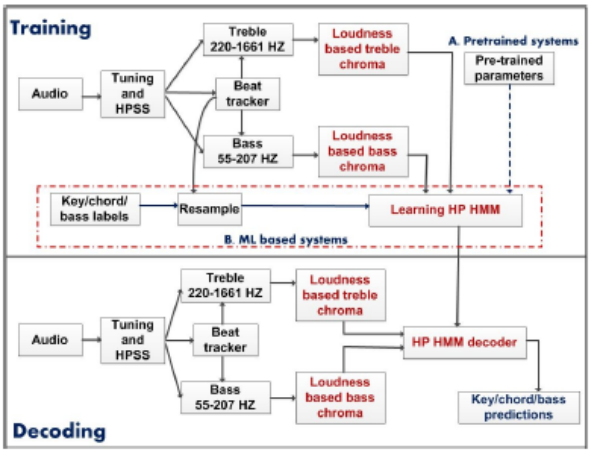

Figure 2: Components of the harmony progression analyzer (HPA). (Taken from [17])

features are: timbre, melody, rhythm, pitch, among others. Depending on the task at hand, some features will be more useful than others. For instance, if the task is musical instrument recognition or genre detection, a timbre related feature, such as Mel-frequency Cepstrum Coefficients, will be more useful than a rhythm feature.

A good feature vector in this case is one that can represent audio structure properly while also having low dimensionality and robustness to harmonic changes.

2.1. Chords

Chords are mid-level musical features which concisely describe the harmonic content of a piece. This is evidenced by chord sequences often being sufficient for musicians to play together in an unrehearsed situation [16].

Ni, Mcvicar, Santos-Rodriguez, and De Bie (2012) proposed a new machine based system called Harmony Progression Analyzer (HPA).

HPA is a machine learning system for the harmonic analysis of popular musical audio. It is focused on chord estimation, although the proposed system additionally estimates the key sequence and bass notes [17].

2.2. Query Engine

After the features have been extracted we need to represent the extracted features in a modeling

framework. The chosen model can be Gaussian Mixtures, Histograms or Topic models, and the model is implicitly related with the audio matching stage [11, 8, 5]. Afterwards, the received query also needs to have its features modeled so that we can move to the audio matching stage where we compare the query against the database.

2.3. Audio Matching

Audio Matching is the stage where we compare the query and database given a certain feature and representation of choice. Since we are interested in comparing sequences of different lengths, the queries are usually short fragments and songs are comprised of multiple fragments. This stage becomes one of solving a subsequence audio matching problem. The comparison is made through a distance measure where we calculate the similarity between a query and a song. In the literature there are various algorithms proposed to handle the subsequence matching, most of them use dynamic programming, such as Dynamic Time Warping [21], SPRING [20], Time Warped Longest Common Subsequence (TWLCS) [7]. The purpose of this stage is to automatically retrieve and rank all songs that musically correspond to a query from all the audio documents contained in the database.

A typical approach in global matching is to calculate the euclidean distance between the query and the songs but other more meaningful approaches dependent on the modeling can also be used, such as Fisher's distance [14].

The main idea of Dynamic Time Warping (DTW) is to align two sequences without the restriction of having same length and find point-to-point alignment that minimizes the error between the two sequences. In DTW, an individual element of one sequence can be matched with at least one and possibly more elements of the other sequence, thus allowing for each sequences to be stretched locally along the time axis. This method is usually computed by dynamic programming. The dynamic time warping cost $D(i, j)$ is defined as follows:

$$D(0, 0) = 0. \tag{1}$$

$$D(0, j) = \infty. \tag{2}$$

$$D(i, 0) = \infty. \tag{3}$$

$$D(i, j) = d(i, j) + min \begin{cases} D(i, j-1) \\ D(i-1, j) \\ D(i-1, j-1) \end{cases} \tag{4}$$

$$\forall (i = 1, ..., |Q|; j = 1, ..., |X|)\ D(Q, X) = D(|Q|, |X|). \tag{5}$$

Notice that $d(i, j)$ is the Lp norm difference of i and j.

By placing an additional constraint, which narrows down the set of positions in one sequence that can be matched with a specific position in the other sequence we obtain the Constrained DTW (cDTW). Given a warping width w, the constraint is defined as follows:

$$D(i, j) = \infty \text{ IF } |i - j| > w \tag{6}$$

cDTW has been shown to be significantly more efficient than DTW for full sequence matching and to also produce more meaningful matching scores.

2.4. Music Retrieval using Gaussian Mixture Modeling

A Gaussian Mixture Model (GMM) is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. GMM estimates probability density as the weighted sum of these simpler Gaussian densities, called components of the mixture. They are often used for clustering. The clusters are assigned by the component that maximizes the posterior probability, and like k-Means, GMM uses an iterative algorithm.

The Gaussian Mixture Model is defined in Eq. 7

$$p(F_t) = \sum_{m=1}^{M} c_m N(Ft, \mu_m, \tau_m). \tag{7}$$

where F_t is the feature vector observed at time t, N is a Gaussian pdf with mean μ_m, covariance matrix τ_m, and c_m is a mixture coefficient.

GMMs of Mel-Frequency Cepstrum Coefficients have been widely used and researched in MIR [2, 1, 8, 3, 5]. Since this approach has been widely researched, its strong and weak aspects are well known. In terms of high-level descriptions of music signals, such as genre, mood or computing timbre similarity between songs, the GMM approach is the most predominant paradigm having led to some success. However, latest research by Aucouturier and Pachet shows this approach has a performance glass-ceiling for polyphonic timbre similarity at around 70% precision even after exhaustive fine-tuning or applying delta-coefficients or Markov modeling. One possible cause for this glass-ceiling in precision is the existence of hubs, false positives which are mostly the same songs regardless of the query, so hubs are songs which are irrelevantly close to all other songs. Further research regarding these hubs has been done by Aucouturier, Defreville, and Pachet (2007) and Pachet (2008).

2.5. Music Retrieval using Histograms

Bag-of-words models can be generically defined as an orderless document representation given certain

frequencies of words from a dictionary, in other words, a sparse vector of occurrence counts of words. These models are characterized by having two main steps.

After performing a feature extraction step, where features of choice, such as Chroma or MFCC are extracted, the feature representation step is performed. Each music is abstracted by several vectors called feature descriptors. A good descriptor should have the ability to be robust to changes up to some extent.

Vector quantization or codebook generation consists in converting vector represented features of a song to audio-words, which also produces a "codebook". An audio-word can be considered as a representative of several similar feature vectors. Typically, the simplest method is performing k-Means clustering over all the vectors. Audio-words are then defined as the centers of the learned clusters. The number of the clusters is the codebook size. Finally, each song can be represented by the histogram of the audio-words, given the centers of the learned clusters and audio-word frequencies, this final step is named, histogram construction.

Bag-of-words have been widely applied and researched in text domain obtaining good results[14, 13]. It also has been applied to the music domain with good results [19, 22, 15].

Lebanon et al. created a representation of the bag-of-words where it is possible to have a continuous and differentiable sequential representation [14]. This representation goes beyond the traditional bag-of-words representation and its n-gram extensions by being able to capture sequential information, and yet it is efficient and effective. It is called locally weighted bag-of-words (LOWBOW).

The smoothing method employed in the traditional bag-of-words model is categorical rather than temporal since no time information is preserved. And temporal smoothing has far greater potential than categorical smoothing since a word can be smoothed out to varying degrees depending on the temporal difference between the two document positions. With this in mind, the main idea behind the LOWBOW is to use a local smoothing kernel to smooth the original word sequence temporally, by borrowing the presence of a word at a certain location in the document to a neighboring location but discounting its contribution depending on the temporal distance between the two locations. To handle the problem that several words can occupy one location through temporal smoothing of words, the authors provided a broader definition of a document [14] ultimately resulting in an association between a document location with a local histogram or a point in the simplex.

The multinomial simplex $\mathbb{P}_m$ for $m > 0$ is the m-dimensional subset of $\mathbb{R}^{m+1}$ of all probability vectors or histograms over $m+1$ objects

$$\mathbb{P}_m = \theta \in \mathbb{R}^{m+1} : \forall i \theta_i \geq 0, \sum_{j=1}^{m+1} \theta_j = 1. \quad (8)$$

Its connection to the multinomial distribution is that every $\theta \in \mathbb{P}_m$ corresponds to a multinomial distribution over $m+1$ items.

The topological structure of $\mathbb{P}_m$, which determines the notions of convergence and continuity, is naturally inherited from the standard topological structure of the embedding space $\mathbb{R}^{m+1}$. The geometrical structure of $\mathbb{P}_m$ that determines the notions of distance, angle, and curvature is determined by a local inner product $g\theta(.,.), \theta \in \mathbb{P}_m$, called the Riemannian metric.

The authors proved theorems regarding the LOWBOW, specifically

Theorem 1. *The LOWBOW representation is a continuous and differentiable parameterized curve in the simplex, in both the Euclidean and the Fisher geometry.*

Theorem 2. *Let $K_{\mu,\sigma}$ be a smoothing kernel such that when $\sigma \to \infty$, $K_{\mu,\sigma}(x)$ is constant in μ, x. Then for $\sigma \to \infty$, the LOWBOW curve $\gamma(y)$ degenerates into a single point corresponding to the bow representation.*

Theorem 3. *The LOWBOW curve $\gamma(y)$ satisfies $||\gamma_\mu(y) - \gamma_\tau(y)||_2 \leq |\mu - \tau| O(K), \forall \mu, \tau \in [0,1]$.*

Where O(K) is a Lipschitz constant. As a result of the lowbow curve being Lipschitz continuous, the curve complexity is connected with the shape and scale of the kernel. Thus, we can represent LOWBOW in a finite dimensional space by sampling the path at representative points $\mu_1, ..., \mu_l \in [0,1]$.

Given a Riemannian metric g on the simplex, its product form

$$g'_\theta(u,v) = \int_0^l g_{\theta(t)}(u(t), v(t))dt \quad (9)$$

defines a corresponding metric on LOWBOW curves. This results in geometric structures that are compatible with the base metric g, such as distance or curvature. For example, the distance between LOWBOW representations of two word sequences $\gamma(y), \gamma(z) \in \mathbb{P}_m^{[0,1]}$ is the average distance between the corresponding time coordinates

$$d(\gamma(y), \gamma(z)) = \int_0^1 d(\gamma_\mu(y), \gamma_\mu(z))d\mu \quad (10)$$

The integrated distance formula in Eq. 10 allows the possibility to adapt distance-based algorithms to the LOWBOW representation. To use a

distance-based algorithm with LOWBOW we just need to replace its standar distance such as the Euclidean distance with LOWBOW's integrated distance or its discretized version.

In summary, this representation generalizes bag-of-words by considering the collection of local word histograms throughout the document. In contrast to n-grams, which keep track of frequently occurring patterns independent of their positions, LOWBOW keeps track of changes in the word histogram as it sweeps through the document from beginning to end [14]. In contrast to n-gram, LOWBOW captures topical trends, and incorporates long range information. On the other hand, it is possible to combine the two, so the LOWBOW is orthogonal to n-gram [13]. The ability to capture topical trends and keep track of sequential information is by having the LOWBOW curves.

LOWBOW is represented employing smooth curves in the multinomial simplex. With this representation there are interesting geometrical features to be used in modeling, applied to tasks of retrieval, classification, filtering, segmentation, and visualization. The distance between LOWBOW curves can be used in various modeling tasks, such as K-nearest neighbors, SVM, or even constructing generative models. Other geometrical features can be used, such as the instantaneous direction of the curve which describes sequential topic trends and their change.The authors showed several applications of the LOWBOW framework, such as to text classification with nearest neighbors or support vector machines, text segmentation tasks and applied Dynamic Time Warping of LOWBOW curves.

The LOWBOW framework achieved good results in practice for the referred experiments. And it has also achieved good results in the scope of video however it has still not been applied in MIR.

2.6. Music Retrieval using Topic Models

Topic Models are a suite of algorithms capable of uncovering the hidden thematic structure of large and unstructured collections of documents. The structure uncovered by topic models can be used to explore an otherwise unorganized collection. For example, rather than finding documents through keyword search alone, we might first find the theme that we are interested in, and then examine the documents related to that theme. Topic Modeling algorithms discover not only the themes but also how those themes are connected to each other, and how they change over time. Topic models have been adapted to many kinds of data from its origin in text, to image and audio domain [4].

The simplest and most intuitive topic model is the Latent Dirichlet Allocation (LDA). LDA represents documents as mixtures of topics. Each topic has probabilities of generating various words intuitively related to the topic. LDA makes three assumptions. One is the bag of words assumption, that the order of the words in the document does not matter. Another is the order of documents does not matter. The third assumption is that the number of topics is assumed to be known and fixed. With this assumptions LDA is still a powerful tool for discovering and exploring the hidden thematic structure. However by relaxing the assumptions of LDA it can easily be used in more complicated models.

As previously mentioned, topic models have been applied in the audio domain, specifically for audio information retrieval purposes [9, 10, 11, 18, 12]. They have been shown to be useful in discovering hidden topics for audio similarity retrieval surpassing other known approaches, such as Histograms and GMMs and have already been applied in a problem of query-by-audio as can be seen in the related work described in this section.

3. Implementation

A baseline, generically speaking, is a measurement of some sort used as a basis for comparison. If we want to validate, study and improve previous work, one possible way of doing this is to have a basis for comparison, a baseline. Regarding this thesis, we decided to use a baseline for numerous reasons. The first reason is we are adapting an already existing technique in text to the audio domain. Creating a complex system from scratch would prove very troublesome due to its complexity. The second reason relates to the study and experimentation of the system: this means we wanted a fully functional system in order to test, study and possibly improve it. Another important aspect regarding baseline study is the ability to reproduce the baseline results.

The audio matching task poses a concern in finding a good baseline because in the literature we could not find any available dataset for audio matching purposes to be used. The authors tend to create their own datasets.

We decided to replicate an audio Bag-of-words approach similar to the one proposed by [19]. The motive to replicate the audio Bag-of-words approach is that we use the lowbow framework which extends bag-of-words by locally weighting the terms with temporal smoothing. Given their similarities and the possibility to also use lowbow as a bag-of-words, by altering a parameter, we can efficiently use bag-of-words in lowbow as a baseline and assert our findings of using a locally weighted bag-of-words instead of a typical bag-of-words.

Due to the unavailability of audio matching datasets we decided to use our own datasets and experiment audio matching in two contexts, with a short-query dataset and covers dataset.

3.1. Feature Extraction

Taking into account the results from our feature study we used Harmony Progression Analyzer for our feature extraction since we want robustness and high level correlation with the harmonic progression.

3.2. Query Engine

We chose the lowbow framework as our query engine because of its novelty in the audio domain and also being capable of retaining sequential information which makes it suitable for our query by example system since it will allow the capture of trends in tempo, articulation and phrasing.

We adopted the lowbow framework by implementing it in C++. This framework extends the bag-of-words by performing temporal smoothing where for each time instant the words are locally weighted, effectively becoming a locally weighted bag-of-words and capturing sequential information, such as local trends whereas the bag-of-words is not capable of doing since it is orderless.

Our audio lowbow framework has the following definitions:

Definition 1. *A song x of length N is a function $x : 1, ..., N \times V \to [0, 1]$ such that*
$\sum_{j \in V} x(i, j) = 1 \ \forall i \in 1, ..., N.$
The set of songs (of all lengths) is denoted by X.

For a song $x \in X$ the value $x(i, j)$ represents the weight of the audio word $j \in V$ at location i. Since the weights sum to one at any location we can consider Definition 1 as providing a local audio word histogram or distribution associated with each song position. The standard way to represent an audio word sequence as a song in X is to have each location host the appropriate single audio word with constant weight, which corresponds to the δ_c representation defined below with $c = 0$.

Definition 2. *the standard representation $\delta_c(y) \in X$, where $c \geq 0$, of an audio word sequence $y = \langle y1, ..., y_N \rangle$*

$$\delta_c(y)(i, j) = \begin{cases} \frac{c}{1+c|V|} & y_i \neq j \\ \frac{1+c}{1+c|V|} & y_i = j \end{cases}. \quad (11)$$

Equation 11 is consistent with Definition 1 since $\sum_{j \in V} \delta_c(y)(i, j) = \frac{1+c|V|}{1+c|V|} = 1$. The parameter c in the above definition injects categorical smoothing to avoid zero counts in the δ_c, representation.

Definition 1 lets several audio words occupy the same location by smoothing the influence of audio words y_j across different song positions. Doing so is central in converting the discrete-time standard representation to a continuous representation that is much more convenient for modeling and analysis.

Definition 1 is problematic since according to it, two songs of different lengths are considered as fundamentally different objects. To allow a unified treatment and comparison of songs of arbitrary lengths we map the set $1, ..., N$ to a continuous canonical interval, which we chose to be $[0, 1]$.

Definition 3. *A length-normalized song x is a function $x : [0, 1] \times V \to [0, 1]$ such that*
$\sum_{j \in V} x(t, j) = 1, \forall t \in [0, 1].$
The set of length-normalized songs is denoted X'

A simple way of converting a song $x \in X$ to a length-normalized song $x' \in X'$ is expressed by the length-normalization function defined below.

Definition 4. *The length-normalization of a song $x \in X$ to a length-normalized song $x' \in X'$ is the mapping*
$\varphi : X \to X' \ \varphi(x)(t, j) = x(\lceil tN \rceil, j)$
where $\lceil r \rceil$ is the smallest integer greater than or equal to r.

The length-normalization process abstracts away from the actual song length and focuses on the sequential variations within the song relative to its length. In other words, we treat two songs with similar sequential contents but different lengths in a similar fashion. For example the two songs $\langle y1, y2, ..., y_N \rangle$ and $\langle y1, y1, y2, y2..., y_N, y_N \rangle$ would be mapped to the same length-normalized representation.

We formally define bag-of-audio-words as the integral of length-normalized songs with respect to time. This definition is equivalent to the popular definition of the traditional bag-of-words.

Definition 5. *The bag-of-audio-words or boaw representation of a song y is $\rho(\varphi(\delta_c(y)))$ defined by*

$$\rho : X' \to \mathbb{P}_{V-1} \text{where} [\rho(x)]_j = \int_0^1 x(t, j) dt, \quad (12)$$

and $[\cdot]_j$ denotes the j-th component of a vector.

Above, $\mathbb{P}_{V-1}$ stands for the multinomial simplex.

Definition 6. *The locally weighted bag-of-audio-words or lowbow representation of the audio word sequence y is $\gamma(y) = \gamma_\mu \in [0, 1]$ where $\gamma_\mu(y) \in \mathbb{P}_{V-1}$ is the local audio word histogram at μ defined by*

$$[\gamma_\mu(y)]_j = \int_0^1 \varphi(\delta_c(y))(t, j) K_{\mu,\sigma}(t) dt. \quad (13)$$

Equation 13 indeed associates a song location with a local histogram or a point in the simplex $\mathbb{P}_{V-1}$ since

$$\sum_{j\in V}[\gamma_\mu(y)]_j = \sum_{j\in V}\int_0^1 \varphi(\delta_c(y))(t,j)K_{\mu,\sigma}(t)dt = \int_0^1 K_{\mu,\sigma}(t)\sum_{j\in V}\varphi(\delta_c(y))(t,j)dt = \int_0^1 K_{\mu,\sigma}(t)\cdot 1dt = 1.$$

The Simplex of the lowbow framework is initialized with the obtained vocabulary from the HPA chords. Query and database songs received are initialized as smooth curves through the corresponding audiowords file receiving categorical and temporal smoothing.

The temporal smoothing in our framework is done through the Gaussian probability density function (pdf) restricted to [0,1] and renormalized:

$$K_{\mu,\sigma}(x) = \begin{cases} \dfrac{N(x;\mu,\sigma)}{\phi((1-\mu)/\sigma)-\phi(-\mu/\sigma)} & x\in[0,1] \\ 0 & x\notin[0,1] \end{cases} \tag{14}$$

where $N(x;\mu,\sigma)$ is the Gaussian pdf with mean μ and variance σ^2 and ϕ is the cumulative distribution function (cdf) of $N(x;0,1)$.

3.3. Audio Matching

The primary reason behind choosing our audio matching algorithm is in regards to what similarity we want to tolerate. As stated previously we are interested in tempo, phrasing and articulation so chord length deviations and insertions/removals are considered. Secondly, it has to be easily adaptable to our subsequence matching algorithm. Thirdly, the time constraint, although it is not the focus of this thesis, we want to obtain an efficient query-by-example system.

Comparing all of the state-of-the-art algorithms presented in section 2.3 the constrained DTW (cDTW) is the best for our system. It captures the similarities in tempo, phrasing and articulation. It also is easily adaptable to subsequence matching, such is the case of SPRING. Additionally it has a better performance than the other algorithms.

We adapted cDTW for our task by using sink states as explained in the SPRING algorithm. We allow the match of a query to start at any point between the start and end of the music being compared while also allowing the match of a query to end at any point between the start and end of the music being compared. We also set a threshold of distance equal to 1 in order to exclude resulting sequences of higher distance which deviate from our query in order to be in accordance with our task. With these changes we effectively solve the query to subsequence problem of our task. We are able to compare between queries and full songs using cDTW to capture the similarities in tempo, phrasing and articulation.

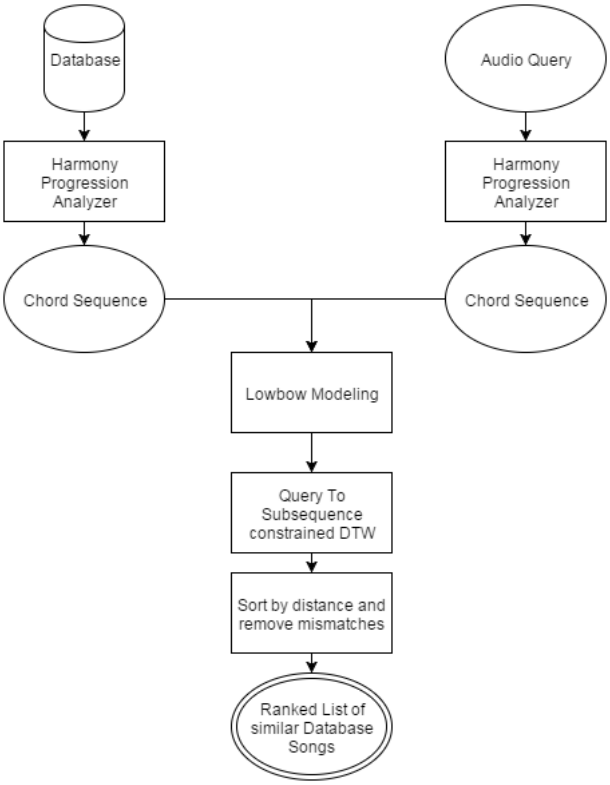

Figure 3: Architecture of our Query-by-Example system.

4. Results

We applied the same settings specified for our feature extraction with Harmony Progression Analyzer, respectively 11025Hz sample rate, 55 minimum frequency and maximum of 207 for the bass chromagram and 220 minimum frequency and maximum of 1661 for the treble chromagram. For our modeling using locally weighted bag-of-words we chose for our sigma $\sigma = e^{-3}$, a smoothing coefficient of e^{-2} and the sampling was done with a 1.0 ratio.

4.1. Short Query Dataset

The dataset is comprised of 700 noise songs and 25 popular guitar songs from various genres, such as blues, rock, metal, etc. For each song we extracted four short length queries between 10 to 15 seconds resulting in 100 queries.

The experiment consisted of running each query against our query-by-example system and retrieving the respective song from which we extracted the short query.

The main purpose of this experiment was to validate our query-by-example system while also demonstrating that lowbow retains sequential information being leveraged in our system.

The metrics used to assess our performance were:
- Hitting ratio $Top(N) = \frac{Hit(N)}{Total}$.

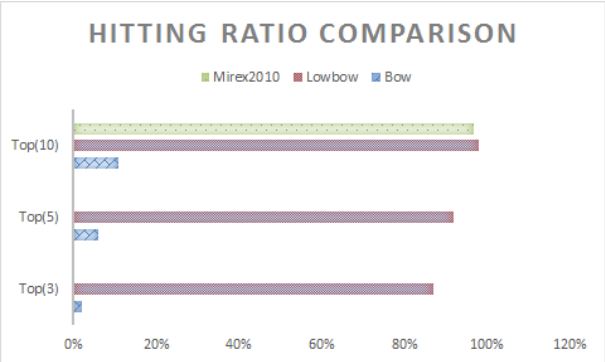

Figure 4: Hitting ratio comparison between Lowbow and Bow.

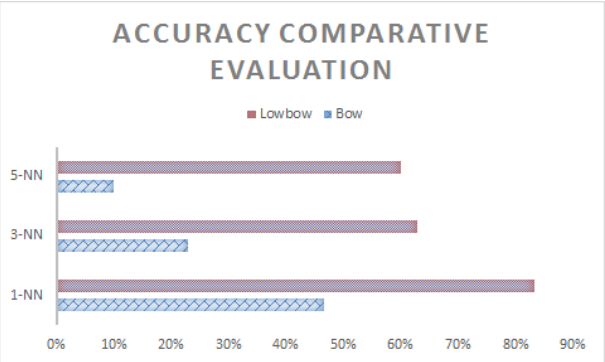

Figure 5: Cover Songs Accuracy Comparative Evaluation.

- Mean Reciprocal Rank $MRR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{Rank(i)}$.

These metrics are also used by the MIREX community for query-by-humming systems considered as standard metrics for query system performance evaluation.

The results are shown in detail in Figure 4.

As we can see in Figure 4, our query-by-example system using lowbow is capable of achieving hitting ratios similar to MIREX's query-by-humming state of the art, a MIREX 2010 submission.

Assessing the results from our task point of view, of retrieving songs where we can play the submitted query. We found that the songs, between the first rank and our target song from which we extracted the query, had higher similarity due to lowbow's smoothing. By removing the smoothing process we observed all these songs had the same DTW distance, respectively 0.

The smoothed curves affect the local points in a way that the musical trends near that point are also retained. For example, a song which repeats the query sequence often, means the trend at the obtained subsequence is highly correlated with the original query whereas a song that has the original query but then shifts to a completely different theme does not.

Model	Mean Reciprocal Rank
Bow	0,03
Lowbow	0,756
Mirex 2010	0,947

Table 2: Mean Reciprocal Rank comparison.

In Table 2 the MIREX 2010 submission of the query-by-humming state of the art outperforms our system, although in the top 10 hitting ratio we have higher accuracy. This result directly translates to the reasoning that for short queries with popular chord sequences it is hard to find at rank 1 the song you are expecting since there are plenty of songs with that popular chord subsequence.

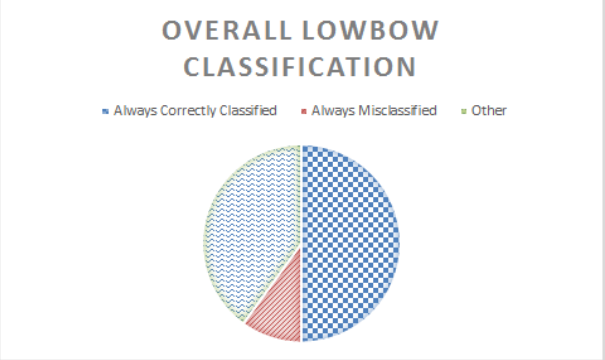

Figure 6: Overall Classification Tendencies of Lowbow.

In terms of our bow baseline comparison, for our matching problem, bow is not capable of finding target songs for our task, as it only has a Mean Reciprocal Rank of 3%.

The lowbow greatly outperforms the bag-of-words baseline. This result can be explained by the sequential information being captured in lowbow which provides excellent results in our query to subsequence matching problem.

4.2. Cover Song Dataset

The cover song dataset is comprised of 700 noise songs, 30 original songs, each one with 10 respective cover songs.

The experiment consisted of running each original song against our query-by-example system and retrieving the respective cover songs. In this experiment we did not use the constraints, therefore allowing cDTW to start and end at any point since we are using full songs.

The results are shown in detail in Figures 5, 6, 7 and a Table 3 of the classification tendencies of bow and lowbow.

As we can see in Figure 5, our query-by-example

Model	Always Correctly Classified	Always Misclassified	Others
Bow	10%	40%	50%
Lowbow	50%	10%	40%

Table 3: Classification Tendencies Lowbow vs Bow

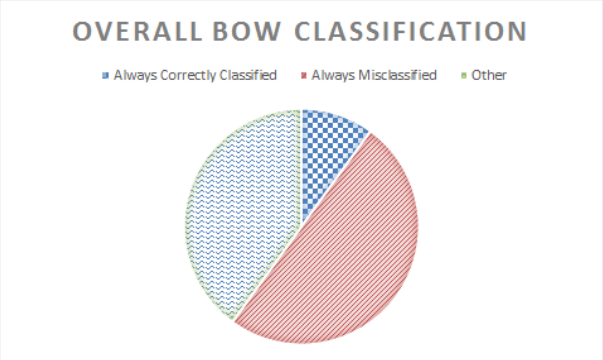

Figure 7: Overall Classification Tendencies of Bow

system using lowbow is capable of detecting cover songs to a certain degree. In accordance with our system characteristics, Figure 6 shows that our system is capable of correctly identifying cover songs which are based on the original song harmonic sequence, whereas cover songs based on melody or rhythm have more harmonic differences.

Assessing the results against our bow baseline, we see that for cover song detection bow is a valid choice for cover songs such as studio to live versions without large harmonic differences as it had already been applied to in related work in chapter **??**.

Lowbow greatly outperforms the baseline as it is capable of extending the range of identified cover songs. This result is clearly noticed by looking at Table 3. These results can be explained again by the sequential information being captured which allows us to tolerate deviances in the harmonic sequence whereas the bag-of-words only takes into account the global harmony without any order.

5. Conclusions

After an extensive study we conclude that our query-by-example system successfully addresses our task of finding thematically similar songs allowing the user to play the query to be submitted and that our system can also be used for certain cover songs detection.

We believe that the future of audio matching will pass by a mixture of sequential information with some symbolic data or supervised learning. The reasoning behind our belief is that audio similarity requires a deeper connection between the sequential information and the musical context in which the sequence occurs. Knowing if we are in the presence of a chorus or a verse will help in improving audio matching and enhance the knowledge obtained from sequential information. With the added information we believe audio matching can grow to something bigger and become capable of relating an original with its cover songs with high precision.

For our particular problem of audio matching in a query-by-example scenario two major objectives must be fulfilled: firstly, a way to evaluate the problem; and secondly, an algorithm to solve it.

Regarding this thesis, we had some difficulties in solving the first problem since audio matching has some inherent subjectivity and there were no available audio matching datasets. Our solution was to validate short queries by creating our own audio matching dataset and also by applying our query-by-example system to the closest audio similarity tasks, cover song detection. The second problem was also one of this thesis goals: the creation of an audio matching query by example system. We solved this problem by using the lowbow framework and studying related work to define an audio bag-of-words baseline from it.

We believe that changing our lowbow framework to capture local contexts, such as song chorus, verses, among others, to be of high value towards improving our system. With these labels users can explicitly tag queries to be of a particular context, e.g. chorus. Afterwards our system would compare that particular query only to chorus sections.

Acknowledgements

I want to thank my advisor, Prof. David Matos, for his sage advice and insightful criticisms.

I am also thankful to my family and girlfriend for being kind and supporting.

References

[1] 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*, pages 1–8, 2002.

[2] J.-J. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *ISMIR*, 2002.

[3] J.-J. Aucouturier and F. Pachet. A scale-free distribution of false positives for a large class of audio similarity measures. *Pattern Recognition*, 41(1):272–284, 2008.

[4] D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

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

[6] P. Grosche, M. Müller, and J. Serrà. Audio content-based music retrieval. *Dagstuhl Follow-Ups*, 3, 2012.

[7] A. Guo and H. Siegelmann. Time-warped longest common subsequence algorithm for music retrieval. 2004.

[8] M. Helén and T. Virtanen. Query by example of audio signals using euclidean distance between gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, pages I–225. IEEE, 2007.

[9] M. D. Hoffman, D. M. Blei, and P. R. Cook. Content-based musical similarity computation using the hierarchical dirichlet process. In *ISMIR*, pages 349–354, 2008.

[10] D. J. Hu. Latent dirichlet allocation for text, images, and music. *University of California, San Diego. Retrieved April*, 26:2013, 2009.

[11] P. Hu, W. Liu, W. Jiang, and Z. Yang. Latent topic model for audio retrieval. *Pattern Recognition*, 47(3):1138–1143, 2014.

[12] S. Kim, S. Narayanan, and S. Sundaram. Acoustic topic model for audio information retrieval. In *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA'09. IEEE Workshop on*, pages 37–40. IEEE, 2009.

[13] G. Lebanon. Sequential document representations and simplicial curves. *arXiv preprint arXiv:1206.6858*, 2012.

[14] G. Lebanon, Y. Mao, and J. V. Dillon. The locally weighted bag of words framework for document representation. *Journal of Machine Learning Research*, 8(10):2405–2441, 2007.

[15] Y. Lu and J. E. Cabrera. Large scale similar song retrieval using beat-aligned chroma patch codebook with location verification. In *SIGMAP*, pages 208–214, 2012.

[16] M. McVicar, R. Santos-Rodríguez, Y. Ni, and T. De Bie. Automatic chord estimation from audio: A review of the state of the art. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22(2):556–575, 2014.

[17] Y. Ni, M. Mcvicar, R. Santos-Rodriguez, and T. De Bie. Harmony progression analyzer for mirex 2011. *Proceedings of the 6th Music Information Retrieval Evaluation eXchange (MIREX)*, pages 1–4, 2011.

[18] L. Ren, D. B. Dunson, and L. Carin. The dynamic hierarchical dirichlet process. In *Proceedings of the 25th international conference on Machine learning*, pages 824–831. ACM, 2008.

[19] M. Riley, E. Heinen, and J. Ghosh. A text retrieval approach to content-based audio retrieval. In *Int. Symp. on Music Information Retrieval (ISMIR)*, pages 295–300, 2008.

[20] Y. Sakurai, C. Faloutsos, and M. Yamamuro. Stream monitoring under the time warping distance. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 1046–1055. IEEE, 2007.

[21] M. Shokoohi-Yekta, J. Wang, and E. Keogh. On the non-trivial generalization of dynamic time warping to the multi-dimensional case. In *Data Mining. Proceeding of the 2015 International Conference on*, pages 39–48. SIAM, 2015.

[22] X. Zhu. Persistent homology: An introduction and a new text representation for natural language processing. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1953–1959. AAAI Press, 2013.