Evaluating the Potential of Similarity Estimation Using Hyperloglog for Query by Humming Problem

Arthur Lubambo

A Thesis Submitted in Partial Fulfilment

of the requirements for the

Degree of

Master of Science in Data Analytics

Diagram

Description automatically generated with medium confidence

February 2023

Supervisor: Kislay Raj

# Abstract

To Do

# Introduction

## Query by Humming

The Query by Humming (QBH) problem is a significant challenge in the area of Music Information Retrieval (MIR), which consists of retrieving or identifying a piece of music based on a user's vocal singing or humming of a melody. This problem encapsulates the difficulty of transforming a non-verbal, subjective representation of a musical piece into a format suitable for comparison and retrieval within a computational framework.

QBH has an inherent complexity due to the variability of vocalizations reproduced by humans. Different from text-based queries, where keywords usually have good matching with the corpus of documents, the humming or vocals do not have by default a standardized notation or representation that is a good match with the songs in the database, to identify and retrieve the musical piece searched in the database. The natural variations of pitch, rhythm, tempo, and timbre reproduced by users make it particularly a hard problem.

This problem was originally studied by (Asif Ghias et al., 1995) who introduced a pioneering approach that utilised the UP, DOWN, and SAME notation  (for example, SUDUDDSUD) for representing the pitch movements of hummed melodies. This encoding was associated with an algorithm for string matching that tolerates mismatches to develop the core of an engine that retrieves a melody MIDI stored in the database that matches best with the humming.

The literature review will cover methods studied by other researchers in the area using a wide range of strategies such as dynamic time warping (DTW), Locality Sensitive Hashing (LSH), and others. There are different contexts and variants of this problem where each method might have a strength or weakness. A common bottleneck between the techniques is the computational resources involved in executing the information retrieval, in time and memory complexity.

With this context mentioned above this project aims to study the potential of Hyperlolog++ Data Structure usage in a method to calculate the similarity between query and songs for QBH problem.

## Cardinality Estimation

The cardinality is the measurement of the size of a set, in other words, it refers to the number of distinct elements in a set or a population. The computation of cardinality is relevant in multiple domains. Depending on the context it might be impractical to calculate the exact solution, due to the size of the dataset, or the nature of the distributed data across multiple nodes. Under this scenario, it was developed solutions that give up an exact result for controlled approximated estimation, using probabilistic data structures getting significant efficiency gains in memory and time complexity to execute the estimation. The literature review covers in more detail the different algorithms for cardinality estimation. It was chosen the Hyperloglog++ as an algorithm in this project due to its superior accuracy, efficiency, and versatility. Its advanced techniques and optimizations make it a preferred choice for many applications requiring fast and accurate estimation of set cardinalities, particularly in the context of large-scale datasets and memory-constrained environments.

## Research Objectives

This project assessed the potential of using Hyperloglog++ in the QBH problem. Associating each Query to a Hyperloglog instance (HLL) and each Song in the Database to another HLL, then computing the similarity by an overlap coefficient, using the estimation of the intersection size, using HLL operations.

### Develop a Hyperloglog-based Similarity Estimation Framework for Query by Humming

It consists of creating sets of encodings for queries and songs then using the jaccard overlapping index estimation based on Hyperloglog++ (HLL) data structure, developing a framework for HLL parameters tuning for query by humming problem.

### Investigate the relationship between memory usage vs performance in Hyperloglog++ based Similarity Estimation

it includes running experiments to discover the impact of memory reduction and performance, establishing a modelling for the relationship between the hyper-parameters and accuracy. Analysing the hyper-parameters impact on memory usage. Finetuning it will bring the key value for the proposed method and its benefits.

### Compare and contrast Hyperloglog++ based Similarity Estimation Framework with other algorithms' performance by metrics.

it includes experimentation and comparison about the weakness and strongness of each type of algorithm used in the experiments under multiple aspects beyond memory usage, and performance, but also analyses in what musical context they tend to work better or not.

## Hypotheses

**Hypothesis 1:** The memory usage of HLL-based algorithms is smaller than the exact set-based solution.

**Hypothesis 2:** The Mean Reciprocal Ranking (MRR) of HLL-based algorithms is equal to or greater than 90% of the MRR from the equivalent exact set-based solution.

## Scope and Limitations

The project focused on the QBH problem, so the algorithm developed is not appropriate for plagiarism detection. The dataset and techniques developed and studied are suitable for the Western music context (12 notes system), so it is not adequate to use it for microtonal music pitch less music, or other types of music. The proposed algorithm requires a discrete musical notation to fit in a set modelling. The project applied different variants of algorithms with these characteristics to analyse the HLL estimation version versus the exact Set version cardinality computation.

## Report Overview

This project Chapter X….. TO DO

# Literature Review

## Query By Humming

Query by Humming (QBH) is an area of Music Information Retrieval that studies the content-based search in the song database. Humming serves as a natural and efficient method for searching a musical audio database by vocalizing the melody of a song. It was originally studied by (Asif Ghias et al., 1995), using the pitch movements “Up”,” Down”, and “Same”, to encode the query, and substring searching in a database of 183 songs was conducted using parameters to accept a controlled level of mismatch. The big challenge at that time was encoding the audio signal in a string (‘U’, ‘D’, ‘S’), due to computational power and algorithms available, taking up to several minutes to process 10 seconds of audio. This Research opens the area of QBH it was quite interesting how a simple method using a UDS encoding could produce good results.

Later, other approaches were introduced to this problem such as dynamic time warping (DTW) a technique used in Times Series, but it could be modelled specifically for the audio processing context. For example (Fu et al., 2007) used the pitch level over time to model it as a times series using the dynamic programming technique DTW to calculate the matching level. Even with the optimizations, it has a quadratic cost to calculate the match between two series where n is the length of the series.

(Tripathy et al., 2009) they followed a very similar approach to the original (Asif Ghias et al., 1995) it created a layer for Wav to MIDI from the query and then used an algorithm for string matching, but they used a different method using dynamic programming to calculate the edition distance between strings.

In general, all the methods try different approaches to deal with the mismatching between what the song is, with the humming sung by a human. It can be observed that it is not only a

time alignment between the query and the song. It is also a pitch alignment problem as was highlighted by (Stasiak, 2012), who proposed a method inspired by auto-adaptive human behaviour for ignoring errors in sung melodies.

All the algorithms were focused on the accuracy of the method so far, but there is another fundamental aspect to enable the implementation of it in practical applications, scalability. It is a natural question, to understand how to apply it in a large database. (Guo et al., 2013) Introduced the application of Locality Sensitive Hashing (LSH), a technique that creates an index based on hashes operations considering the similarity between the content. They had applied this method developing other layers to tackle the problem of key transposing, it is when the music and the query are not in the same tone. So it was more scalable and improved the performance with better mean reciprocal rank (MRR).

(Alexios Kotsifakos et al., 2011) proposed a subsequence matching framework capable of dealing with gaps either in the query or the target song, performing better than the other Dynamic Programming methods, and maintaining the same time complexity.

Considering the diversity of queries, (Wang and Jyh-Shing Roger Jang, 2015) started to extract other type information from the queries that were not being used before, the lyrics. It is quite common besides humming the people to sing part of the song, or even whistle. So this study explored how the usage of lyrics in addition to the melody might impact the performance of QBH. They have combined speech recognition techniques to extract the lyrics of the humming combining it with the melody distance, reducing the error rates significantly. For obvious reasons, it would just impact music and queries with lyrics. The other challenge of this method is the language context, to make it generalizable it is necessary to have multiple languages trained, and lyrics structured for all the songs. So it implies extra effort in its implementation.

A common problem with the songs is the complex extraction of the melody in an automated way. The original studies were based on existing MIDI databases with the songs already in the symbolic format. Producing it algorithmically through raw audio format is more complex, the song usually the song has multiple instruments, with harmony and a combination of more than one melody at the same time. With this challenge in mind, (Alfaro-Paredes, Alfaro-Carrasco and Ugarte, 2021) used a voice separation to improve the melody extraction from the songs, it demonstrated better results for the encoded melody for the song, consequently improving the matching algorithms.

Most of the studies were trying to use improvements in the string match algorithm to consequently have better results with the QBH, but (Velankar and Parag, 2018) changed the paradigm, introducing the matching using n-grams and inverted index.

They combined it with the Mean Normalized Frequency (MNF) Algorithm and developed a method called “Unified Algorithm for Melodic Music Similarity”. The big advantage of this method is that it tackles the information retrieval reducing the search space for the songs where the n-Grams matches. So, the pre-computed indexes help the reduction of the query time. The N-Gram contains each segment of intervals for the song, so even with imperfections on pieces of the query encoding is unlikely to compromise the result.

As a counterpoint to (Velankar and Parag, 2018) the work from (Ulfi and Mandala, 2022) highlighted that the “Unified Algorithm” has issues with performance and works slow for big datasets. They also implement the algorithm “Query by Humming System using Frequency-Temporal Attention Network” but apply enhancements to the Partial Matching of queries on it. In the end, it concludes the method has problems with scaling for big datasets. It might indicate a gap to be explored by applying adjustments in the algorithms for scalability efficiency.

In General, all of the algorithms are based on two areas: melody extraction from the song and query; and melody matching. All of them had a different approach to encode the melody, to extract it, or to match it. Some of the methods also tackled the search problem, creating an index to retrieve the relevant candidates more efficiently. The encoding format of the melody varied between two categories: Discrete Sequence, or Continuous Sequence. Some other researchers explored multi-media format, using lyrics in addition to the typical format.

### Melody Extraction

The melody extraction or Transcription is the name of a task that transforms the musical audio signal in a temporal pitch notation with the sequence of notes. As mentioned before, it is an essential step for the QBH algorithms.

The most recent application of Machine Learning techniques on it shifts the usage from fully algorithmic solutions to data-driven models. For instance, (Yu et al., 2021) used a method inspired by the human perception of frequency, time and intensity for audio, applying Convolutional Neural Networks to extract the melody. Or (Donahue, Thickstun and Liang, 2022) use generative models to create the synthetic dataset and combine it with annotated datasets to improve the melody transcription. Or (Bittner et al., 2022) who proposed a lightweight neural network for musical transcription, it is compatible with polyphonic outputs and could be applied to a wide range of instruments and vocals.  
  
We could see some of the methods, explicitly or implicitly used source separation for better results in the QBH, It is especially important as a pre-processing layer for the melody extraction phase. (Défossez, 2022) used a combination of Frequency Domain with Time Domain, introducing the method “Hybrid Spectrogram and Waveform Source Separation”, it is based on deep neural networks architecture.

The advantage under-explored from Source Separation for QBH is the ability to encode more than one simultaneous melody. It might improve the multi-vocal, contrapoint songs. allowing the matching with the query to be on any of the melodic lines.

### Dataset

The dataset for QBH might be a challenge to produce, it needs to be done with a carefully designed procedure, to avoid bias introduction. It has been found (Salamon, Serrà and Gómez., 2012) a database with 118 recordings of sung melodies, used in multiples studies from this Literature Review. It was built from 17 subjects, keeping a good gender balance and a wide level of music knowledge level, from zero to amateur musicians. They were presented with a list of songs, and they were free to pick the ones they knew and were asked to sing any part of the melody for recording. There was no restriction on time, or what part of the melody must be sung, they were free to sing with or without lyrics. They didn’t listen to the original song before recording it. All the records were made from a simple microphone from a laptop in order to simulate a realistic scenario for QBH. This dataset is robust given all the experimentation designs and the wide usage in academic research.

## Probabilistic Algorithms and Data Structures

In QBH a big challenge is to enable more accurate results, with algorithms that scale well for big datasets. For instance, Spotify Dataset has more than 100 Million Tracks (Spotify, 2023). It is necessary to have more efficient methods for QBH. This Project will explore the potential of Probabilistic Algorithms and Data Structures in the QBH Task.

One of the most impactful data structures created was the HyperLogLog. (Flajolet et al., 2007) did enhancements in the original loglog algorithm for the cardinality estimator, making it possible to estimate the size of sets beyond 10^9 with a standard error of 2% while using a memory of only 1.5 kilobytes. This data structure creates multiple registers of hashes and computes operations of add, count and merge (two HLLs). The extremely low memory has a cost with the accuracy, but it can be tunned by a parameter that increases the number of registers. The application of this Data Structure in Big Data was disruptive, making it possible the count distinct elements estimation in a distributed system extremely quickly. (Heule, Nunkesser and Hall, 2013) made improvements to the method increasing the accuracy and reducing even more memory usage.

As mentioned previously, there were some methods that used locality-sensitive hashing LSM (Guo et al., 2013) (Matti Ryynänen and Anssi Klapuri, 2008) and MinHash to index the songs for QBH. Although there are improvements, in the MinHash algorithm called HyperMinHash (Yun William Yu and Weber, 2020), there is a good opportunity to investigate if this new implementation would impact the performance of the QBH.

Also, one of the applications of HLL could be inclusion coefficient estimation as (Nazi et al., 2018) demonstrate in an efficient method applied in database columns schema analysis. This coefficient could be defined as the fraction of the intersection of two sets relative to the smaller set. This concept can be used as an inspiration for another context. For example in the QBH, the matching between the query and the song can be modelled as the inclusion coefficient index between the melodic encodings of the query and song, using it as a set of keys. So the challenge is to transform into how to encode it with relevant information for matching it. In addition to it, the HyperLogLog methods could be used to optimize the execution, creating a scalable implementation for it.

# 

# Proposed primary research methodology

The primary research methodology employed in this study centred around experimental investigations. The main objective is to evidence the cause-and-effect relationship between the application of the proposed model and the query-by-humming performance metrics. The goal is to demonstrate that enhancements in the algorithm's execution did not affect the query-by-humming performance negatively.

In many other areas, this type of data structure is demonstrating improvements in the tasks. The Literature Review demonstrated a lack of studies with the HyperLogLog applied to Query By Humming, so this research aims to explore this approach and evaluate the results. The methodological framework was structured across 3 phases: data preparation, implementation utilizing the proposed approach with fine-tuning and experimentation, and results analysis comparing the new framework against alternative methods.

A screenshot of a computer

Description automatically generated

Fig 1 Diagram with Research methodology

The initial stage of the Research will be identifying suitable secondary datasets for the primary research of this capstone project. It is a key area, It is important to choose reliable sources, with representativeness, and good usage in other studies. Then the dataset will be cleaned and prepared for the next phases. The cleaning consists of standardising the song identifiers between the query dataset identification and song identification. It will enable the data to be crossed between both datasets.

Then the melodic extraction for the song dataset will be executed, It is very important because the audio tracks from the songs might include multiple instruments, and vocals, so the algorithm needs to mark from all simultaneous pitches which of them represents the main melody.. Then the Wav to MIDI algorithm will transcribe the pitches of the melody for both datasets. Alternatively, in case the challenges of finding a suitable raw WAV dataset with the songs contained, it can be used an existing symbolic song dataset with the Midi transcriptions already done. The focus of this research is not the melodic extraction phase of the algorithm, instead, it is the encoding and retrieval part of songs in the database using HLL hashes.

With the Midi transcriptions ready, two operations for Relative Pitches Sequence and Rhythmical Sequences extraction need to be applied to create the Discrete Features Set for each song or query. The relative pitches are important because each person might sing in a different note the same melody, also the rhythmic encoding must be robust to different speeds of the song. Note this algorithm could be extended to any type of high-dimensional discrete feature sequence for a song.

Then the dataset needs to be prepared for the experiment, splitting into multiple folds for using a cross-validation strategy, finishing the Data Preparation Stage.

The Experimental Phase: Prepare each model candidate for the analysis, including multiple variants of the proposed method similarity estimator based on HLL, Overlapping Coefficient and other alternatives such as dynamic time warping (DTW), Scaling and Time Warping (SWM). Each model will be executed multiple times for each Fold.

Each execution will be evaluated against memory usage, execution time, accuracy, and Mean Reciprocal Rank (MRR). After collecting the results, the data visualization of the experiment results will be implemented, and the statistical analysis with the Hypotheses Test will be executed. The comparison will be done by highlighting the takeaways from the experiment results. The first key comparison will be between different hyper-parameters of the model, and the second key comparison, will be the same model but using the typical Set Cardinality estimation instead of HLL. Then at last compare with the other algorithms of the same problem.